POSSIBLE APPLICATION OF QUALITY FUNCTION DEPLOYMENT IN SOFTWARE SYSTEMS DEVELOPMENT IN THE UNITED STATES AIR FORCE

THESIS

Craig R. Lamb, Captain, USAF

AFIT/GSS/LSY/91D-8

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSS/LSY/91D-8

POSSIBLE APPLICATION OF QUALITY FUNCTION DEPLOYMENT IN SOFTWARE SYSTEMS DEVELOPMENT IN THE UNITED STATES AIR FORCE

THESIS

Craig R. Lamb, Captain, USAF

AFIT/GSS/LSY/91D-8

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GSS/LSY/91D-8

POSSIBLE APPLICATION OF QUALITY FUNCTION DEPLOYMENT IN SOFTWARE SYSTEMS DEVELOPMENT IN THE UNITED STATES AIR FORCE

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Software Systems Management

Craig R. Lamb, B.S.E.E.
Captain, USAF

December 1991

Preface

The purpose of this research project was to determine if the techniques of Quality Function Deployment (QFD) could be applied to software development in the United States Air Force environment, and whether or not further effort should be expended in this area. The ideas and concepts used in the more common hardware application of QFD, were modified to apply to software development. This research was limited to the early requirements analysis phase, although Software QFD can be used throughout the software development process. If you are interested in QFD I would encourage you to contact either the American Suppliers Institute or the GOAL/QPC organization. Both organizations are non-profit groups focusing on studying and teaching the techniques of Japanese total quality management.

In conducting this research project and writing this thesis I have had a great deal of help from others. I would like to thank Dr Ben Williams for unknowingly providing the initial inspiration for this thesis. Dr Williams also provided the support and funding necessary to procure the QFD Designer software which proved invaluable in completing this project. I would also like to thank Mr Mark Miller of QualiSoft Corporation for his cooperation in acquiring QFD Designer. I also wish to thank Mr Allen Chartier of the American Suppliers Institute for his help in identifying sources of valuable information on QFD. My thanks also extends to the various people who took the time to study and respond to my rather lengthy survey. Finally, I am especially grateful to my thesis advisor, Lieutenant Colonel Chris Arnold, who provided guidance and support mixed with the proper application of motivational techniques needed to keep me focused these last ten months.

Captain Craig R. Lamb

Table of Contents

Page

Page

List of Figures

Page

List of Tables

Abstract

The objectives of this thesis were to determine whether the methods of Quality Function Deployment (QFD) could be used in the software development environment within the USAF, and whether or not this area should be researched further. The research was limited to the requirements analysis and definition phase.

The different areas of study included a brief review of the structured analysis methodology, a detailed review of the QFD models currently being used in the product industries, a review of how QFD fits into the software development cycle, and specific software modifications to the QFD methodology. A review of some applications of software QFD (SQFD) is also performed. A sample problem consisting of an Automated Teller Machine is developed in detail using the SQFD methods identified.

A subjective survey was conducted of a small sample of USAF software experts to determine the suitability of SQFD for USAF use based on the sample problem.

The results of the thesis show that QFD can be adapted to software development. SQFD also shows potential to save both time and money for the USAF. Consideration must be given to the sample size and nature of the survey when interpreting the results of this research.

POSSIBLE APPLICATION OF QUALITY FUNCTION DEPLOYMENT IN SOFTWARE SYSTEMS DEVELOPMENT IN THE UNITED STATES AIR FORCE

I. Introduction

Background

Software development is a difficult and challenging area of weapon systems procurement and has been the focus of much criticism and attention. The development of software has often been characterized by missed schedules, cost overruns, and flawed products. As made clear by many articles on the subject, the "Software Crisis" is upon us. Newer weapons systems incorporate increasing amounts of software. For example, the B-1A strategic bomber contained 500,000 lines of code (Canan, 1986:46). Just ten years later, the B-1B contained 1,200,000 lines of code (Canan, 1986:46). The C-5A contained only 25,000 lines of code as compared to its eventual replacement, the C-17 which will have between 625,000 and 750,000 lines of code (Kitfield, 1989:33). During the recent war in the middle east, many of the systems used employed software to fulfill their mission so successfully. The E-3 Airborne Warning and Control System (AWACS) was designed to track 600 aircraft simultaneously and accomplishes its mission with 515,000 line of code (Richards, 1990:A6). The Aegis cruiser relies on 3.7 million lines of code to perform its mission (Richards, 1990:A6). With the success of these weapon systems and the growing complexity and performance demanded of them, it is evident

that the growth in software size will continue. The software for the Advanced Tactical Fighter may require between 5 and 7 million lines of code (Kitfield, 1989:30). One estimate of the size of the software for Phase 1 of the Strategic Defense Initiative (SDI), including simulation and engineering environment, ranges from 25 million to 40 million lines of code (Myers, 1989:93).

All of this software is not free and as software increases in importance and shear size, so too will its cost. Behind all of the automated machines and computers today is a $125 billion-dollar-a-year industry which influences our banking, airline reservations, telephone networks, and even flushes the toilets on the newest Boeing 747 (Richards, 1990:A-1, A-24). Expenditures for the Department of Defense are also significant and on the rise. In a Memorandum to the Joint Logistics Commanders in 1985, Deputy Secretary of Defense William H. Taft, IV, stated that annual expenditures of mission critical computer software had reached $9 billion in 1985 and was projected to cost $30 billion in 1990 (Committee on Science, Space, and Technology, 1989:1). The Army estimates that the cost of software increases twofold per year when the costs of support systems and logistics are taken into account (Kitfield, 1989:30).

With so much of the resources of society being spent on software, both for its own use and taxpayer dollars spent on defense, even a small percentage of lost funds due to errors are significant. This cost is most likely thought of in terms of dollars, but with the increasing reliance on software human life must also be considered. To reduce the penalty for inefficient software development we must look at the sources of software errors.

The classic approach to software development can be viewed in terms of the Waterfall Model. The Waterfall Model was first described by Royce as early as 1970,

after which numerous refinements and variations have been suggested (Sommerville, 1989:7). The stages of the model can be seen in Figure 1 below.

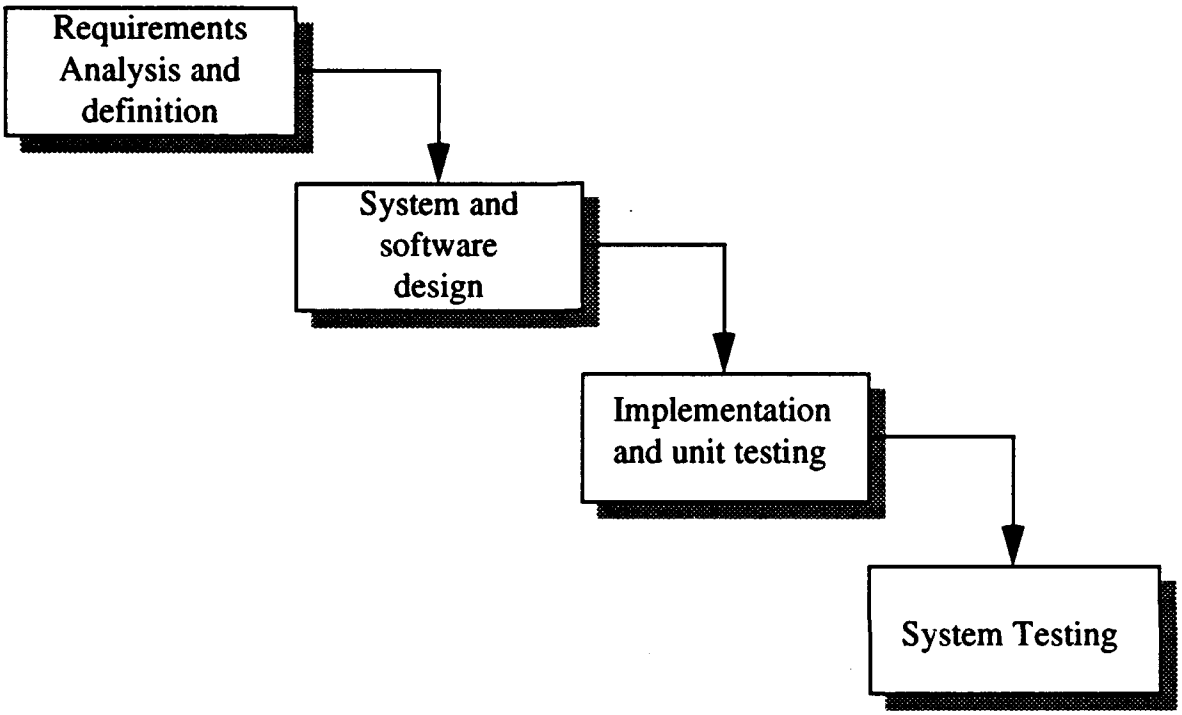


Figure 1. Waterfall Model of Software Development (Sommerville, 1989:8)

The purposes of these stages are:

Requirements Analysis and Definition. The system's services, constraints and goals are established by consultation with system users. Once these have been agreed, they must be defined in a manner which is understandable by both users and development staff. (Sommerville, 1989:7)

System and Software Design. Using the requirements definition as a base, the requirements are partitioned to either hardware or software systems. This process is termed systems design. Software design is the process of representing the functions of each software system in a manner which may readily be transformed to one or more computer programs. (Sommerville, 1989:7)

Implementation and Unit Testing. During this stage, the software design is realized as a set of programs or program units which are written in some

executable programming language. Unit testing involves verifying that each unit meets its specification. (Sommerville, 1989:7)

System Testing. The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer. (Sommerville, 1989:7)

By looking at the Waterfall Model we can see how each phase feeds from the products of the succeeding phase. Unfortunately this is also the path that errors take through the model. One estimate claims that sixty percent of all software errors can be traced to the design phase (Stewart, 1988:49). This is due to inadequately specifying the requirements of the software system. Brooks states, "I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation" (Brooks, 1987:11). Errors made early in the software development cycle can be very expensive to fix. This is born out by some estimates which claim that it can cost 6, 10, and even 100 times as much to correct an error in the maintenance phase (post system testing phase) as it would have cost to correct it in the design phase (Stewart, 1988:48). If this much can be gained through preventing design errors from passing on into systems even more may be gained from preventing requirements errors from becoming design errors to begin with.

Problem Statement

Quality Function Deployment (QFD) has been shown to be of great value to the automotive industry. QFD is an overall concept that provides a means of translating customer requirements into the appropriate requirements for each stage of a product development life cycle (Sullivan, 1986:39). After seven years of experience with QFD at Toyota, a 61% reduction in start-up costs, a one third reduction of the product development cycle (time to market), and fewer design changes overall were realized

(Sullivan, 1986:50). A case study at Eaton Corporation found that the use of QFD to design a blend door actuator for automobiles resulted in: 30% reduction in size, 50% reduction in selling price, 50% reduction in engineering expenses, 20% reduction in drafting expenses, a reduction in noise from 50 decibels to 38 decibels, and mounting flexibility allowing it to be used on three additional car lines (De Vera, 1988:38). If techniques could be developed to apply the QFD process to USAF software development, it is possible that improvements in the software development life cycle could be realized reducing cost, schedule, and design changes.

Investigative Questions

In investigating how Quality Function Deployment (QFD) could be applied to Software Systems Development, several investigative questions (IQ) need to be answered. This research effort will attempt to provide a possible answer to these questions:

- IQ1 - What are the QFD process, its goals, products and techniques?
- IQ2 - What are the current techniques available to develop software systems requirements for the USAF?
- IQ3 - Can the QFD process be tailored to meet a specific domain's requirements?
- IQ4 - What specific tailoring of the QFD process should be made to use QFD to aid in developing software systems requirements in the USAF environment?
- IQ5 - Is the QFD process acceptable to USAF software managers/ engineers?

- IQ6 - Does the application of the tailored QFD process result in requirements analyses that correct shortcomings of current techniques?

Scope

The scope of this research effort will be limited to the application of QFD in the requirements analysis and requirements definition phase of software development efforts as managed by the USAF.

II. Literature Review

Introduction

This chapter contains a review of literature organized into the following topic areas:

- Current techniques available to the USAF for software requirements analysis and definition. The specific methods to be reviewed are those associated with structured analysis. The notations of the structured analysis technique will also be discussed. Examples will be presented using the structured analysis methods to familiarize the reader with this approach.

- The QFD model as described by two mainstream approaches, the Akao model and the Clausing-Makabe model. An example will be presented using both QFD models to familiarize the reader with QFD in general and the two models specifically.

- Current literature and experiences of the use of QFD in the software development environment.

Current Requirements Analysis Methods

This section reviews a current technique for developing software requirements. The technique to be reviewed here is structured analysis as it will be used to develop our sample problem in Chapter IV. In addition to the structured analysis technique the following notations will be discussed: data flow diagrams, data dictionaries, and entity-relationship diagrams.

Requirements Analysis Techniques

As the software engineering process matures the importance of requirements analysis has been realized. Current trends in software systems characteristics indicate that size is increasing rapidly, thereby increasing system complexity, in order to meet demands for greater functionality (Kitfield, 1989:29-33). As systems progressed past the point of being manageable by just one or two people, methods to control the software process were in greater demand (Yeh, 1990:450). Key to building a system that meets the user's needs is requirements analysis (Yeh, 1990:452). The goal of the requirements analysis is to describe *what* a system should do as opposed to describing *how* it should do it (Davis, 1990b:119).

One of the first formally described methods was structured analysis. This method has been widely used in many software projects. It is characterized by a functional approach to a system decomposition and the use of simple and easy to understand notation. Additional methods have been developed as well. Some of these are modifications and/or extensions of the structured analysis method. Object oriented analysis and design has also emerged as a new methodology. We will only study the structured analysis method in sufficient detail to understand the sample problem to be developed later.

Structured Analysis. The ideas of structured design/analysis were first suggested by Larry Constantine in a book based on 16 years of work in the area (Yourdon, 1979:xi-xiii). The goal of structured analysis is to produce a structured specification that provides a concise and easy-to-understand model of a system through a top-down functional decomposition (Martin, 1985:401).

Data Flow Diagrams. Data Flow Diagrams (DFDs) are a fundamental tool in structured analysis. DFDs show, as the name implies, how data flows through the system and is processed through the system. They do not show the order of events or any logical decisions (Davis, 1990a:59). In addition, DFDs should not be constructed with too much or too little detail (Yourdon, 1979:189). Details such as error paths, for example, should not be shown (Yourdon, 1979:189).

The DFD consists of a few easily understood graphical symbols; some of which are shown in Figure 2 below. A named arrow shows the data flow itself. The bubble shows a transform (or process) of data into other data. The terminator, shown as a named rectangle, indicates the source or destination of data. Two parallel lines represent data in static storage. (Davis, 1990a:57)

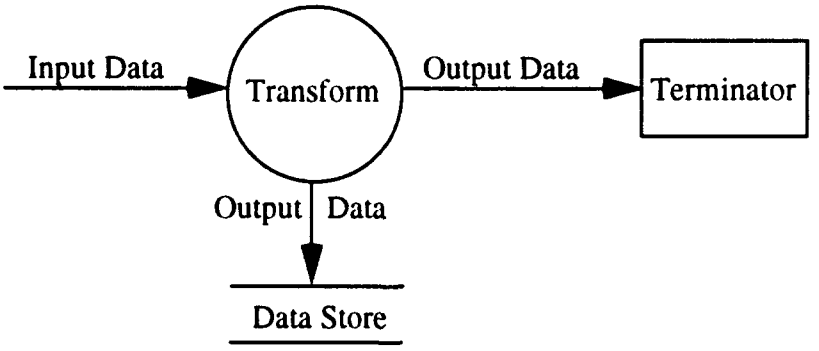


Figure 2. Basic Level 1 Data Flow Diagram (Yourdon, 1979:338; Davis, 1990a:57)

The level 1 DFD may be further defined, or leveled, to allow a large complex system to be represented as subsystems consisting of lower level DFDs (DeMarco, 1978:72). An example level 2 DFD is shown in Figure 3 below. This DFD is one level below the level 1 DFD shown above. When thought of in this way, the top level DFD is called a Context Diagram (DeMarco, 1978:75). The lower level DFDs must have inputs

and outputs equivalent to the parent bubble that it is spawned from (DeMarco, 1978:78). When this equivalence is reached, the DFDs are said to be balanced (DeMarco, 1978:78). In the example level 2 DFD we see that the input data flow and output data flows are the identical data flows of the level 1 DFD (context diagram). The internal data flows (intermediate data flows) and transforms (A, B, and C) in the level 2 DFD show the data flow and transforms that take place inside the level 1 DFD transform bubble.

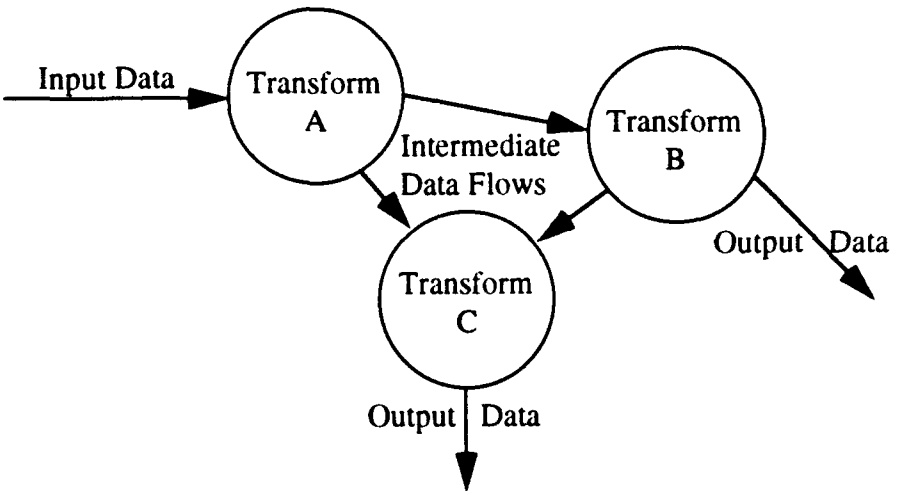


Figure 3. Basic Level 2 Data Flow Diagram

A sample level 1 DFD, considered a context diagram, is shown in Figure 4 below. This DFD represents the simple flow of information in a telephone. The terminator *caller* inputs the *voice* signal and the *keyed phone number* data into the transform *telephone call*. The *telephone call* transform then processes the inputs and results in the *sound* as a data flow into the terminator called *receiver*. (Pressman, 1987:167-168)

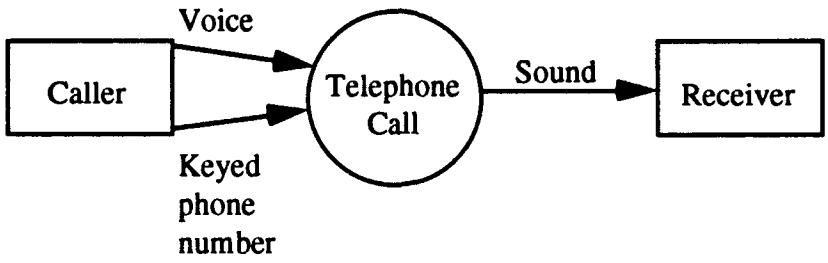


Figure 4. Level 1 Telephone DFD (Pressman, 1987:168)

The above context diagram may be further simplified by leveling it down to a level 2 DFD. The level 2 DFD is shown in Figure 5 below. The two data flows into the system and the one data flow out of the system are identical in the context diagram and the level 2 DFD. The main transform of the context diagram has been further defined into the four transforms shown in the level 2 DFD below. Data flows between these lower level transforms, shown by the named arrows, are internal to this level 2 DFD.

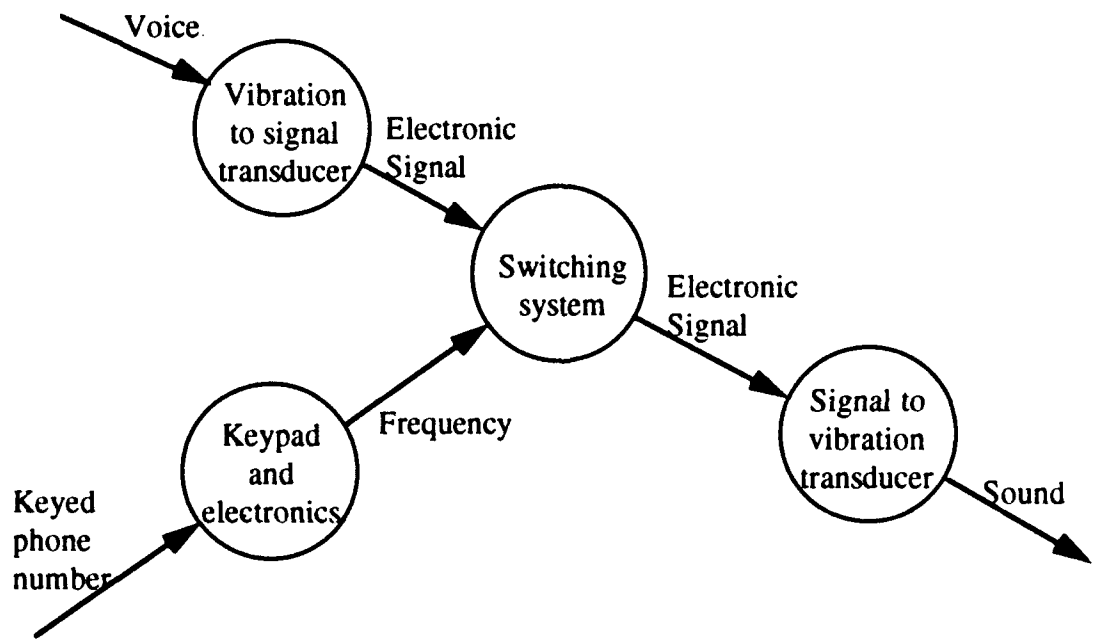


Figure 5. Level 2 Telephone DFD (Pressman, 1987:168)

Data Dictionaries. The data flows contained in the DFDs can be further defined using the notation of the data dictionary. Each data flow, or named arrow, is represented as a type of formal equation describing the make up of the data item. The basic notation is shown in Table 1 below.

Table 1

Data Dictionary Notation (Pressman, 1987:173; Davis, 1990a:62; Peters, 1987:31)

Data Construct	Notation	Meaning
	=	is composed of
Sequence	+	and
Selection	[...|...|...]	selection of one alternative
Repetition	{ }^{n}	n repetitions of
	( )	optional data
	" "	a literal string or value
	* *	comment

Some examples of these data equations for the telephone system DFDs described previously can be seen in the data dictionary in Table 2 below. This data dictionary contains all the definitions associated with *keyed phone number*. A simple entry, *outside number*, is shown to consist of the number *9* plus the *local number* or *long distance number*. According to the notation, either one or the other may be included but not both. Furthermore, the data item *local number* is divided into a *prefix* plus *access number*. the *long distance number* consists of the optional *0* plus *area code* plus *local number*. All

data flows in the system are represented to their lowest composition in the data dictionary (Pressman, 1987:173).

Table 2

Data Dictionary for Keyed Phone Number (Pressman, 1987:174)

Data Flow	Composition
keyed phone number	= [local extension | outside number | 0]
local extension	= [2001 | 2002 | ... | 2999 | conference set]
outside number	= 9 + [local number | long distance number]
local number	= prefix + access number
long distance number	= (0) + area code + local number
conference set	$= \{\# + \text{local extension} + \#(\#)\}_{2}^{6}$

Entity-Relationship Diagrams. Data can also be represented by showing the relationships among data entities. The resulting diagrams are called Entity-Relationship Diagrams or E-R diagrams. "An entity is something, real or abstract, about which we store data" (Martin, 1985:297). Some examples of entities might be *caller*, *sound*, *receiver*, and *local number*.

Relationships exist among entities as represented by the E-R notation. Several different notations exist. Some notations represent entities as rectangles with relationships (IS MADE OF, REFERS TO, CREATES etc) described inside of angled brackets connecting various entities together (Peters, 1987:58-60). Another notation uses

rectangles again for entities but named diamonds to show relationships and attached circles to show attributes (Davis, 1990a:63). Since we are only concerned with a cursory understanding of E-R diagrams, we will briefly look at two notations, Crow's-Foot notation and Arrow notation.

These basic notations are shown in Figure 6 below. Links between entities describe the number of occurrences of one entity to another. In addition to this quantity relationship, a label may be placed on the link between entities to better describe the relationship. (Martin, 1985:315-322)

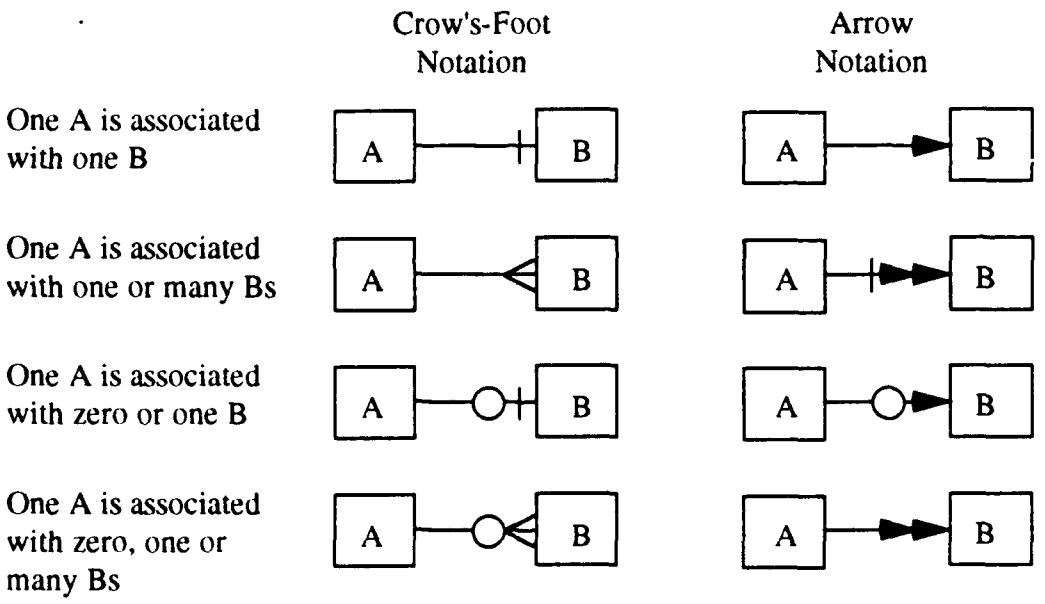


Figure 6. Entity-Relationship Diagram Notation (Martin, 1985:320)

The example Crows'-Foot E-R diagram below (Figure 7) shows the relationships between the entities *Driver*, *Car*, *Truck*, and *Motor Bike*. According to the E-R diagram, the *Car*, *Truck*, and *Motor Bike* can have one, many, or no *Driver*. However, the *Driver* can only be associated with either one *Car*, one *Truck*, or one *Motor Bike* at a time, as shown by the mutually exclusive relationship represented by the filled in dot (Martin, 1985:300).

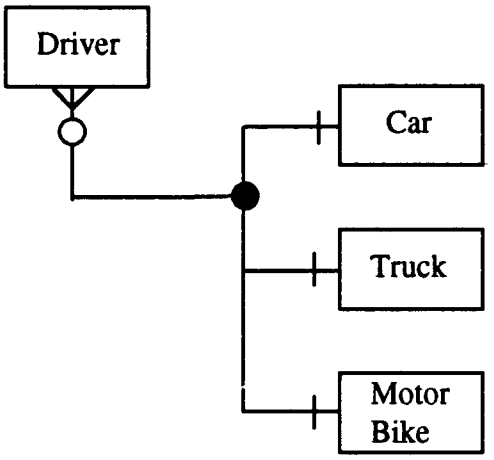


Figure 7. Mutually Exclusive Entity-Relationship Diagram (Martin, 1985:322)

Shortcomings of the Structured Analysis Technique. Structured analysis captures a physical model of the system being developed by the use of DFDs, data dictionaries, and, at times, E-R diagrams. These representations are only as good as the gathering of the requirements used to build them. Gathering these requirements is part of the structured analysis method and is key to a successful system. Many authors have identified the gathering and communicating of user needs as a very important step in the process (Davis, 1990a:20; Peters, 1987:16; Sommerville, 1989:54-55). Part of the problem stems from the differences in the way users (high level of abstraction) and developers (details for specification purposes) describe a particular requirement (Sommerville, 1989:55). A user may require user friendliness, however, the developer may desire the requirement stated such that "all user command selections should take place using command menus" (Sommerville, 1989:55). Methods to better translate these abstract user requirements into more detailed developer requirements may prove beneficial to the software system being developed.

Structured analysis also does not provide any means of prioritizing the users' many and varying requirements. This results in user requirements that only slightly

satisfy the user being treated with the same priority as those which are very important in order to satisfy the user.

Additionally, the structured analysis method does not include any means of tracing the user's requirements through the analysis to ensure they are properly represented in the design. The USAF sees the value in tracking the user's requirements at the top level (Statement of Operational Need and System Operational Requirements Document) and requires a requirements correlation matrix in AFR 57-1 *Operational Needs, Requirements, and Concepts* (Department of the Air Force, 1988:9). In order to ensure that these top level requirements are flowed down into the specifications and ultimately the final product, a means of tracing the requirements from the user into the system must be used.

Quality Function Deployment

Quality Function Deployment (QFD) is a planning process that began in the Mitsubishi Kobe shipyards in 1972 (Hauser, 1988:63). QFD is not solely a quality tool as its name would imply. Quality Function Deployment is taken from six Chinese/Japanese words meaning: *hin shitsu* (qualities, features, or attributes), *ki no* (function), and *ten kai* (deployment, development, or diffusion), which combined, address qualities (i.e. features) and development rather than just quality. (Eureka, 1988:2) Mr George R. Perry, Vice President, Quality and Reliability, Allied-Signal Inc defines it as:

> A systematic way of ensuring that the development of product features, characteristics, and specification, as well as the selection and development of process equipment, methods, and controls are driven by the demands of the customer or marketplace. (Eureka, 1988:2)

Another definition from the founder of QFD, Dr Yoji Akao:

> Quality Function Deployment...provides specific methods for ensuring quality throughout each stage of the product development process, starting with design. In other words, this is a method for developing a design quality aimed at satisfying the consumer and then translating the customers' demands into design targets and major quality assurance points to be used throughout the production stage. (Akao, 1990:3)

So as we can see from these definitions, QFD is a planning process intended to take a product from initial concept though design and production ensuring that the end product meets the needs of the customer. The term "customer" may be used interchangeably with the term "user." The customers' needs play a key role in QFD. As we will see later, it is these needs, properly analyzed, that decide which way the product will evolve. Indeed, the final production plans should be traceable back to the original customer demand. For example, a worker on a production line installs a light-weight part made from light-weight materials in order to meet the customers' demand for a product that *is easy to lift* from the initial QFD planning phase.

QFD is only one of many activities that fit into the larger picture of Japanese management philosophy. This is shown in the following figure of Total Quality Control (Management) from the Japanese perspective (Figure 8). It is important to realize that while QFD may be a powerful planning process, it is not a stand alone solution to poor management. The reader can learn about each of these activities from various sources. Only QFD will be covered in this research paper.

There are two mainstream QFD models currently accepted and used in industry today. The first model to be discused was developed by the founder of the QFD process, Dr Yoji Akao. This model, refined over the years since its inception, is promoted in the United States by the non-profit Growth Opportunity Alliance of Lawrence/Quality-

Productivity-Competitiveness (GOAL/QPC) organization and its Executive Director, Mr Bob King. The Akao model is the more rigorous approach to QFD, utilizing a possible combination of 30 quality matrices (King, 1989:2-6). The second model to be discused was adapted by Dr Don Clausing and Dr Hajime Makabe from the original Akao QFD model. This is a less complex QFD model based on four phases (Eureka, 1988:18). This approach is promoted in the United States by the non-profit group American Suppliers Institute (ASI) and its Vice President and General Manager, Mr Bill Eureka. The models will be addressed separately.

Figure 8. Total Quality Control (TQC/TQM) Wheel (King, 1989:7)

Before exploring these two QFD models, it is important to note that each model should be tailored to the specific task at hand (Eureka, 1988:35). There are differences in the two models as well as similarities, however, neither model is presented here as the predominant or best approach to QFD. The individual strengths and weaknesses can be judged by the reader.

The Akao-GOAL/QPC QFD Model. The interpretation of the Akao model presented here follows the approach proposed by the GOAL/QPC organization as outlined in the King text. This approach is well organized and accepted by many in industry who practice QFD.

This QFD model suggests that QFD can be thought of as broken into four phases: Organization, Descriptive, Breakthrough, and Implementation (King, 1989:2-1).

The organization phase focuses on the subject of the QFD study as defined by management. Additional factors such as the scope of the study, the intended benefactor of the study, schedules, and costs are additionally set forth. Also important are the composition of the QFD team and the goals of the study. (King, 1989:2-2)

The descriptive phase attempts to define the product in terms of the customers' demands, the quality characteristics needed to achieve those demands, the functions the product performs, and its major subgroups and/or parts. In addition, new technologies such as materials are addressed, new ideas or views of the product are brought forth, and the product's failure modes are described. (King, 1989:2-3 to 2-4)

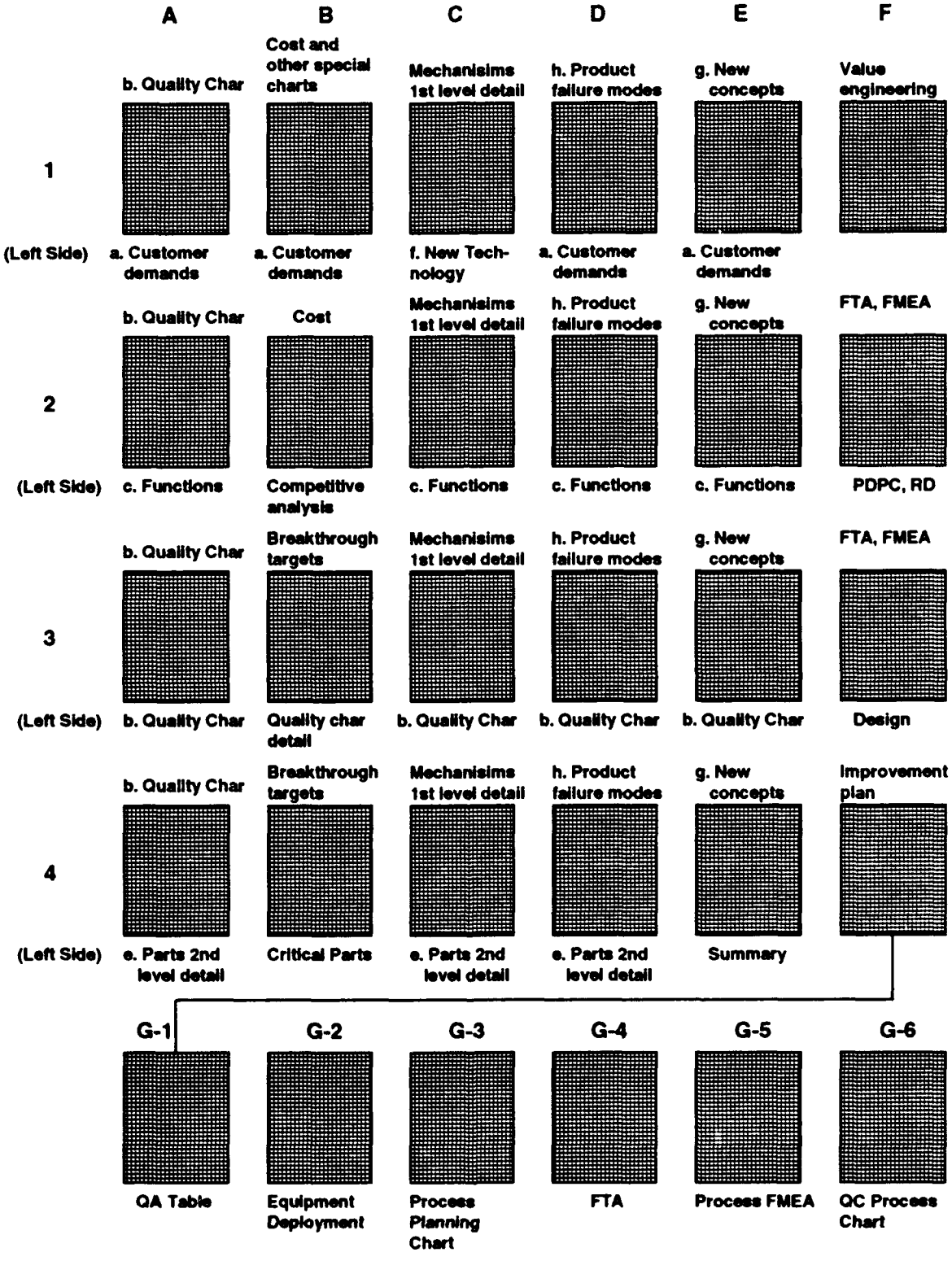


Figure 9. GOAL/QPC Matrix of Matrices (King, 1989:2-6)

In the breakthrough phase the items resulting from the descriptive phase are brought together into a matrix of matrices to view the product from different perspectives in order to achieve new ideas and concepts. This matrix of matrices is shown above (Figure 9). (King, 1989:2-5)

The matrix of matrices shows all 30 possible QFD charts of the GOAL/QPC model. These charts represent the entire life cycle from requirements through production. Each rectangle represents one QFD chart. The labels marked "left side" are the items that appear on the left hand side of the QFD chart, while the items at the top of each rectangle are the items that appear across the top of the QFD chart. For the first chart in the QFD model, chart A-1, the items on the left hand side of the chart would be *Customer Demands* and the items across the top of the chart would be *Quality Characteristics*. Items used more than once in the life cycle are labeled with letters. For example, *Customer Demands* (labeled a) is used in chart A-1, B-1, D-1, and E-1. The matrix of matrices is arranged so that the charts are accomplished from column to column starting with column A (King, 1989:2-6). The final charts to be completed are the G-series charts. Some charts will be discussed in greater detail as a sample problem is developed. Other charts will only be briefly touched upon to acquaint the reader with the overall concept of QFD.

Finally, in the implementation phase the product defined above is designed, readied for production, produced, and sold. The products will have benefited from the planning aspects of the QFD matrices as represented by the relationships among the matrix of matrices. (King, 1989:2-5 to 2-13)

The Quality Table - Chart A-1 Part 1. The first matrix to be completed in the Akao-GOAL/QPC model is the Quality Chart or chart A-1 (Figure 10).

In order to understand the QFD process, we will develop a sample quality chart using a mousetrap as a simple problem. The mousetrap example was tailored from a QFD chart delivered with the QualiSoft Corporation's QFD Designer software (QualiSoft, 1991). The purpose of the quality chart is to guide the initial plans for the new product by relating the customers' demands and the importance of those demands to the customer, to the quality characteristics needed to ensure those demands are meet. The chart also rates the company's abilities and capabilities vice the marketplace competition to identify key areas that will benefit the company the most to develop. (King, 1989:4-1 to 4-10)

The first step in constructing the quality table is to list the customers' demands on the left hand side of the matrix. These are the "Whats" of the customers' demands and are arranged in a hierarchy of three levels. Several different tools can be used to gather the needed information including surveys, face-to-face interviews, customer complaints, and customer experimentation with sample products (King, 1989:4-4). In our example one such hierarchy of these customers' demands are: *no mice*, *effective*, and *foolproof*.

The second step is to rate the importance of the customers' demands. Both models use a scale of one to five with five being the most important (King, 1989:4-5; Akao, 1990:28; Eureka, 1988:35). These ratings can be established through a survey, knowledgeable persons, or may have to be guessed (King, 1989:4-5). For the mousetrap, *effective luring*, *reliability*, *foolproof*, *easy to bait*, and *easy to set* were considered the most important customers' demands.

The next step is to rate the company's current position on each of the customers' demands. Each competitor's current position is listed as well. Both are rated on the same one to five scale. The ratings of the competitor's products may be established through

Customers' Demands (Whats)			Customer Importance	Company Now	Competitor X	Competitor Y	Company Plan	Rate of Improvement	Sales Point	Absolute Quality Weight	Demanded Quality Weight
No mice	Lure	Effective luring	5	2	3	2	3	1.50	1.0	7.50	10.6%
	Effective	Reliable	5	3	4	4	4	1.33	1.0	6.66	9.4%
		Foolproof	5	3	3	5	5	1.66	1.2	10.00	14.1%
		Kills quickly	4	2	4	1	4	2.00	1.5	12.00	17.0%
		Kill signal	4	4	2	1	4	1.00	1.0	4.00	5.6%
		Will not slip	3	3	3	4	3	1.00	1.0	3.00	4.2%
		Right size	3	3	4	2	3	1.00	1.0	3.00	4.2%
	Easy to use	Easy to bait	5	4	3	4	5	1.25	1.2	7.50	10.6%
		Easy to set	5	4	3	5	5	1.25	1.5	9.37	13.2%
		No mess	4	4	3	5	5	1.25	1.5	7.50	10.6%

Figure 10. GOAL/QPC Quality Table (Chart A-1 Part 1)

tests, trade magazines, and/or other means. (King, 1989:4-5 to 4-6) Our company currently does well in: *easy to bait* (four), *easy to set* (four), *no mess* (four), but does poorly in *effective luring* (two) and *kills quickly* (two). Competitor X outperforms us in *kills quickly* (four) but underperforms in *easy to bait* (three), *easy to set* (three), and *no mess* (three), while competitor Y does better in *foolproof* (five), *easy to set* (five), and *no mess* (five).

After the company and its competitors have been rated, the company's planned position for the new product is established. This also uses the same one to five scale as before. This position is based on where the company stands now, where its competitors stand, and the company's overall business plan. A competitor's future position may also be considered. (King, 1989:4-6) Our company plans to improve in the areas of *foolproof*, *easy to bait*, *easy to set*, and *no mess* as these are strong holds for the competition. Each of these customers' demands is assigned a value of five.

The rate of improvement is simply the company's plan divided by the company's current position (King, 1989:4-6). For *easy to bait* the rate of improvement is simply 5 / 4 = 1.25.

The sales point is a means of placing greater emphasis on some of the customer demands. A sales point of 1.5 is assigned to customers' demands that are considered major selling points, 1.2 assigned to those considered a lesser sales point, and 1.0 assigned to all others. Only a few customers' demands should be made major selling points to prevent this factor from becoming meaningless. (King, 1989:4-6) For our mousetrap, *kills quickly*, *easy to bait*, and *no mess* are considered to be major selling points and are assigned a value of 1.5.

Finally, absolute quality weight and demanded quality weight are calculated. The absolute quality weight is calculated by multiplying the rate of importance by the rate of improvement by the sales point. In this example, for *easy to bait*, the absolute quality weight is: 5 x 1.25 x 1.2 = 7.5. The demanded quality weight is calculated by dividing the each item's absolute quality weight by the total of the absolute quality weights Eq (1), or in other words making the absolute quality weight into a percentage.

$$\text{Demanded Quality Weight}_i = \frac{\text{Absolute Quality Weight}_i}{\Sigma \text{ Absolute Quality Weights}} \times 100\% \qquad (1)$$

For our example, the demanded quality weight for *easy to bait* is 7.5 / 70.4 x 100% = 10.6. From the last column the top three to four customers' demands can be singled out for greater emphasis (King, 1989:4-6). *Foolproof, kills quickly*, and *easy to bait* account for a large percentage of the quality demanded by the customers. A gain in these areas will be more beneficial then the same gain in another, less important area.

The Quality Table - Chart A-1 Part 2. The second part of the chart A-1 is then filled in (Figure 11). These are the "Hows" for each of the "Whats" from the customers' demands. These "Hows", or quality characteristics, are the items that need to be developed and controlled to ensure that the customers' demands are met (King, 1989:4-7).

As with the customers' demands, the quality characteristics are listed at the top of the quality table in a hierarchy of three levels. The quality characteristics are developed for each "What" from the customers' demands. The characteristics should be items that are measurable, controllable, and testable, and not names of parts or assembles. (King, 1989:4-7) In our example some of these quality characteristics are: *luring radius, dead mouse ratio, skid resistance*, and *setting force*.

Customers' Demands (Whats)		Quality Characteristics (Hows)	Luring	Killing						Convenience				Customer Importance	Company Now	Competitor X	Competitor Y	Company Plan	Rate of Improvement	Sales Point	Absolute Quality Weight	Demanded Quality Weight
			Luring radius	Size	Dead mouse ratio	Time to kill	Auditory	Visual	Skid resistance	# Sizes	# Usable baits	Setting force	Unlatch effort	Debris radius									
No mice	Lure	Effective luring	◉	○	◉	◉			△	○	◉				5	2	3	2	3	1.50	1.0	7.50	10.6%
	Effective	Reliable			◉	◉			○			○	○		5	3	4	4	4	1.33	1.0	6.66	9.4%
		Foolproof		△	◉	○			○		○	○			5	3	3	5	5	1.66	1.2	10.00	14.1%
		Kills quickly		△	◉	◉	◉					◉			4	2	4	1	4	2.00	1.5	12.00	17.0%
		Kill signal					◉	◉							4	4	2	1	4	1.00	1.0	4.00	5.6%
		Will not slip		△					◉						3	3	3	4	3	1.00	1.0	3.00	4.2%
		Right size	○	◉						◉				△	3	3	4	2	3	1.00	1.0	3.00	4.2%
	Easy to use	Easy to bait	△	△							◉	○			5	4	3	4	5	1.25	1.2	7.50	10.6%
		Easy to set		△							◉	◉			5	4	3	5	5	1.25	1.5	9.37	13.2%
		No mess							○				○	◉	4	4	3	5	5	1.25	1.5	7.50	10.6%
Total Absolute Weighted Quality			84.00	91.37	258.00	198.00	36.00	36.00	107.00	49.50	249.37	264.87	42.50	70.50									
Percent of Total Absolute Weighted Quality			5.6%	6.1%	17.3%	13.3%	2.4%	2.4%	7.1%	3.3%	16.7%	17.8%	2.8%	4.7%									
Units			Feet	Square inches	Percent	Seconds	dB	Feet	Friction coefficient	Number	Number	Newtons	Newtons	...ches									
Company Now			20	8.0	95%	0.05	90	10	0.80	4	10	0.5	0.75	4									
Competitor Y			15	8.5	95%	0.10	87	15	0.85	4	7	0.5	0.70	4									
Competitor X			20	7.5	96%	0.15	85	7	0.90	4	5	0.7	0.60	3									
Company Plan			25	7.5	97%	0.04	90	10	0.85	4	15	0.45	0.75	4									

Figure 11. GOAL/QPC Quality Table (Chart A-1 Part 2)

Once the quality characteristics are filled in across the top of the quality table, a relationship to the customers' demands is placed in the resulting matrix of demands and characteristics. This relationship is shown by one of three symbols, each with its own weighting. The double circle shows a strong relationship and has a weighting of nine, the single circle shows some relationship and has a weighting of three, and the triangle shows a possible relationship and has a weighting of one. If no relationship exists, the matrix is left blank. (King, 1989:4-7) For example, *easy to bait* is strongly related to *number of usable baits* and has some relation to *setting force*. It has no relation to *unlatch effort, debris radius* etc.

After all the relationships have been filled in, the weighting of all the relationships in each column is multiplied by the absolute quality weight for that customers' demand, and the total is placed in the row beneath the matrix (King, 1989:4-7). For *skid resistance* this becomes: 7.50 x 1 (triangle-possible relationship) + 6.66 x 3 (circle-some relationship) + 10.00 x 3 (circle again) + 3.00 x 9 (double circle-strong relationship) + 7.50 x 3 (circle again) = 107.00. This is repeated for each quality characteristic.

Next, a percentage row is established in the same manner as for the Demanded Quality Weight (King, 1989:4-7). Using an equation similar to Eq (1), for *skid resistance* we have: 107.00 / 1487.11 (the sum of the individual weighted quality characteristic) = 0.071 or 7.1%.

Below the weighted quality characteristics row are placed rows for each measurable quality characteristic's current value, for each competitor's value and finally the company's target value (King, 1989:4-8). In our example, *dead mouse ratio* has a current value of 95% while competitors X and Y have values of 96% and 95%

respectively. Since *dead mouse ratio* has a high demanded quality value, our company's plan for this quality characteristic is 97%.

This completes the A-1 chart. By examining the A-1 chart, the top three to four customers' demands can be compared to the top three or four quality characteristics to highlight the areas where the emphasis may be placed in product planning and later phases of product definitization.

Quality Characteristics/Functions - Chart A-2. This QFD matrix compares the quality characteristics and the functions of the product (Figure 12). Where the last matrix incorporates the customers' demands, this matrix concentrates on the engineers' demands.

Functions/Service			Quality Characteristics (Hows)													
			Luring		Killing						Convenience					
			Luring radius	Size	Dead mouse ratio	Time to kill	Signal: Auditory	Signal: Visual	Skid resistance	# Sizes	# Usable baits	Setting force	Unlatch effort	Debris radius
Kills mice	Setting up/down	Latching								○		◉		
		Baiting									◉			
		Placing	◉						○					
		Cleaning											◉	◉
	Operation	Latch trips			◉		◉							
		Lures mouse	◉	△							◉			
		Kills Mouse			◉	◉			○	○				
		Signals Trip					◉	△						

Figure 12. GOAL/QPC Quality Characteristics/Functions (Chart A-2)

The quality characteristics from the A-1 chart are brought down to the top of this chart, and then the functions or services of the product are arranged down the left side of the matrix. Relationships are determined using the same symbols and meanings as in the A-1 matrix. From looking at this completed matrix, it is possible to determine inappropriate functions by a lack of a relationship between a function and a quality characteristic. These functions may be deleted or quality characteristics added as appropriate. The same cross-check may be performed for the quality characteristics by checking for a related function. (King, 1989:5-1 to 5-4) In our example, the function *kills mouse* has a strong relationship to *dead mouse ratio* and *time to kill*, and some relationship to *skid resistance* and *number of sizes*. All functions have related quality characteristics and vice versa.

Quality Characteristics/Quality Characteristics - Chart A-3.

The purpose of this QFD matrix is to show how the quality characteristics relate with one another (Figure 13). It is important to know this in order to see how changing one quality characteristic will impact another quality characteristic. As before the double circles represent a strong positive relationship and the single circle some positive relationship. Two new symbols are used: the # shows strong negative relationship and the X shows some negative relationship. An arrow may also be used in order to show the direction of improvement of the quality characteristic. When there is a target value, and improvement above or below this value is not desired, a dash may be used. Some users of the Taguchi methods may represent a planned experiment by the use of a T as a symbol. (King, 1989:6-1 to 6-4)

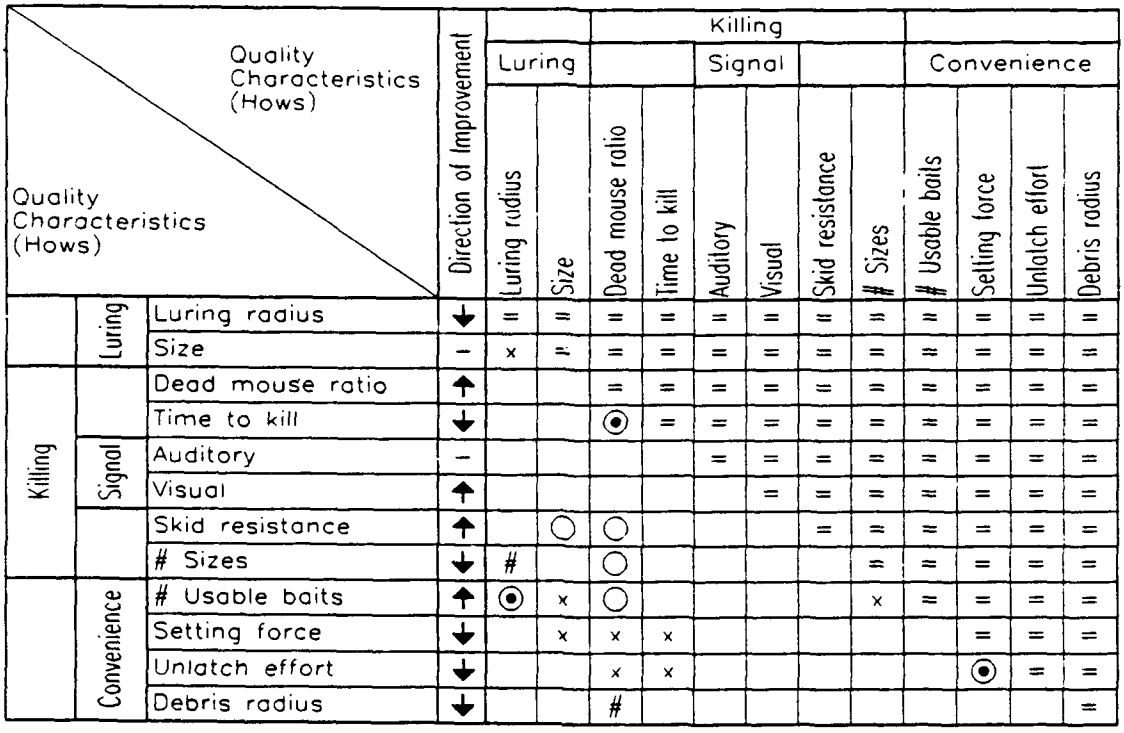


Figure 13. GOAL/QPC Quality Characteristics/Quality Characteristics (Chart A-3)

Following our example, the quality characteristic *dead mouse ratio* has a strong positive relationship to *time to kill.* This tells us that an improvement in one may lead to an improvement in the other. Conversely, *size* has some negative relationship with *number of usable baits.* This tells us that as we change the size of the mousetrap, we must be careful that we do not compromise the number of usable baits. The direction of improvement for each of these quality characteristics is shown by the arrow. *Dead mouse ratio* is improved by increasing the ratio, *time to kill* is improved by decreasing the time, and *size* has no real preference for improvement.

Additional GOAL/QPC Charts. The first three charts presented above are equivalent to the ASI model Phase I chart which will be presented in the next

section. These three charts are only a fraction of the possible charts that may be used with the GOAL/QPC model. In examining the matrix of matrices presented earlier, we see that we have only completed a few of the A column charts. The purpose of the other charts will now be briefly touched upon. Detail is not necessary here since the information represented in these charts is beyond the scope of this research effort. However, it is important for the reader to understand how the entire GOAL/QPC QFD model works in order to see the larger picture of the process.

- ***Chart A-4 Quality Characteristics/Parts.*** This chart contains the most critical quality characteristics across the top and the parts on the left. Relationships are identified in the resulting matrix by the symbols previously discussed. The purpose of this chart is to highlight the critical parts so they may be controlled and optimized. (King, 1989:7-1)

- ***Chart B-1 Functions/Customers' Demands.*** This chart contains the product or service across the top and the customers' demands on the left. Relationships are determined using the established method, and weighted columns are added to determine relative value. These can be combined with cost information to determine targets for value engineering efforts. The purpose of this chart is to identify functions which have an actual cost that exceeds the expected cost. (King, 1989:8-1)

- ***Chart B-2 Cost Deployment Main Chart.*** This is more of a table than a chart. It lists the market price, sales volume, market share, and targeted manufacturing cost as related to current, competitors, and the planned products. The purpose of this chart is a target manufacturing cost for the product. (King, 1989:9-1)

- ***Chart B-3 Quality Characteristics Detail/Breakthrough Targets.*** This chart is also similar to a table. It lists the critical quality characteristics from chart A-1

and the major factors for each characteristic. The level of difficulty may also be shown. The purpose of this chart is to focus effort on the most critical areas for the particular product. (King, 1989:10-1)

- ***Chart B-4 Quality Characteristics Plan/Critical Parts.*** Again this chart is really a table. This table describes the parts which have a strong relationship with the selected quality characteristics from chart A-4. Items of key interest such as its functions, critical quality characteristics, target values, variations allowed, current cost among others are included. The purpose is to identify how the quality and costs of critical parts will be controlled. (King, 1989:11-1)

- ***Chart C-1 Mechanisms/New Technology.*** This chart contains the first level of detail of the product across the top and new technologies on the left side. The relationships are established and displayed in the standard way (i.e., with the symbols previously discussed). The purpose is to identify new technologies which may be of benefit and result in new opportunities. (King, 1989:12-1)

- ***Chart C-2 Mechanisms/Functions.*** This chart has the mechanisms from chart C-1 across the top and the functions from chart A-2 on the left. The relationships are established and displayed in the standard way. Cost information is also included across the bottom. The purpose of this chart is to identify the mechanisms and the functions they relate to, as well as potential areas for cost reduction. (King, 1989:13-1)

- ***Chart C-3 Mechanisms/Quality Characteristics.*** This chart contains the mechanisms from chart C-1 across the top and the quality characteristics from chart B-3 on the left. The relationships are established and displayed in the standard way. The expected value of each quality characteristic is placed in a column on the right side of the

chart. The purpose of this chart is to identify which mechanisms relate to the critical quality characteristics in order to focus effort in these areas. (King, 1989:14-1)

- ***Chart C-4 Mechanisms/Parts.*** This chart has the mechanisms from chart C-1 across the top and the full list of parts from chart A-4 on the left. The relationships are established and displayed in the standard way. The value of the part as a percentage of the whole is listed in a column on the right. The actual cost of the parts is also listed. The purpose is therefore to identify cost contribution of each part. (King, 1989:15-1)

- ***Chart D-1 Product Failure Modes/Customers' Demands.*** This chart has the product failure modes (generated from a fault tree analysis) across the top and the customers' demands from chart A-1 on the left. The relationships are established and displayed in the standard way. The bottom of the chart represents the weight of the product failure modes for each possible failure mode. The purpose of this chart is then to prioritize these failure modes, as related to the customers' demands, based on this weight. (King, 1989:16-1)

- ***Chart D-2 Product Failure Modes/Functions.*** This chart is similar to chart D-1 except that it compares product failure modes with the functions from chart A-2. A weight is again established so that the failure modes may be prioritized from a standpoint of the functions rather than the customers' demands. (King, 1989:17-1)

- ***Chart D-3 Product Failure Modes/Quality Characteristics.*** Again a chart similar to chart D-1 but this time with the quality characteristics from chart A-1 on the left. A weight is again established so that the failure modes may be prioritized from a standpoint of the quality characteristics rather than the customers' demands or functions. (King, 1989:18-1)

❑ ***Chart D-4 Parts Failure Modes/Parts.*** This chart is similar to the last three, however, the part failure modes are placed across the top and the parts from chart A-4 are listed on the left. A column is added to the right which lists the appropriate Failure Mode Effects Analysis (FMEA) study number. The resulting weight is used to prioritize the FMEA studies based on the most critical failure parts. (King, 1989:19-1)

❑ ***Chart E-1 New Concept Selection/Customers' Demands.*** This chart is a type of matrix. Across the top are new concepts to be considered and the customers' demands from chart A-1 are listed on the left. An additional column is listed down the middle which represents the current standard for each customers' demand. This is also referred to as the best in class. A plus or minus is entered into the matrix to show whether the new concept is better or worse than the standard. The columns are totaled, and the results show which new concepts are of value and which are not from the standpoint of the customers' demands. (King, 1989:20-1)

❑ ***Chart E-2 New Concept Selection/Functions.*** This chart is similar to chart E-1 except that the product functions from chart A-2 are listed on the left. The standards for each function are listed in the middle. The totaled pluses and minuses show which new concepts are of value and which are not from the standpoint of the product's functions. (King, 1989:21-1)

❑ ***Chart E-3 New Concept Selection/Quality Characteristics.*** Again a similar chart to E-1 and E-2 except this charts looks at the quality characteristics from chart A-1. The totaled pluses and minuses of this chart show which new concepts are of value and which are not from the standpoint of the quality characteristics. (King, 1989:22-1)

❑ ***Chart E-4 New Concept Selection Totals.*** This chart is the summary of the last three charts. The totals from the first three E charts is brought forward and an overall total is determined for the new concepts. This can help identify which new concepts to proceed with and which to leave. (King, 1989:23-3)

❑ ***Chart F-1 Cost Breakthrough/Value Engineering.*** This is not a chart or matrix but is really a mini-project in itself. The information gathered from charts B-1, C-2, C-3, and C-4 is analyzed to determine the value of functions, quality characteristics, mechanisms, and parts. Several phases are followed in this project:

❑❑ Organization Phase - identifying the proper people, definition of the project, focus of the project.

❑❑ Information Phase - the organization of the available information in order to determine the function value weight.

❑❑ Innovation Phase - using techniques such as brainstorming and nominal group technique to identify breakthroughs for the areas of greatest benefit identified above.

❑❑ Evaluation Phase - a sort of sanity check on the above ideas.

❑❑ Implementation Phase - selling the ideas that have been developed to key decision makers.

All these phases are aimed at achieving the desired performance at the overall lowest cost. (King, 1989:24-1 to 24-6)

❑ ***Chart F-2 Fault Tree Analysis and Reliability Breakthroughs.*** This again is not really a chart or a table but a classic Fault Tree Analysis (FTA). The

purpose is to identify the root cause of failures instead of the symptoms. These can then be managed to increase the reliability of the product. (King, 1989:25-1 to 25-2)

❑ ***Chart F-3 Design Engineering Breakthroughs/ Reviewed Dendrograms, Factor Analysis.*** This is really more a phase than a chart. In an attempt to organize the effort of achieving design engineering breakthroughs, this phase looks at dendrograms and factor analysis.

❑❑ Dendrograms are hierarchically arranged charts which show relationships between parent and child members of the chart. This chart takes a tree-like form. An example would be a chart which shows the position of man in the animal kingdom:

Kingdom	*Animal*
Phylum	*Vertebrate*
Class	*Mammals*
Order	*Primates*
Family	*Hominidae*
Genus	*Homo*
Species	*Sapiens*

In this phase, a dendrogram is constructed of the various new ideas being considered. Each idea is compared to a set of criteria such as durability, cost, effectiveness, regulations etc. Each item in the dendrogram is then compared to these criteria with a question followed by an answer. For example, "Does this increase cost?", "Yes, cost will increase sharply." This technique can highlight areas which may require further testing, and also provides an historical record of rational for each breakthrough item.

❑❑ Factor analysis is carried out on the topics identified from the dendrograms. Factor analysis attempts to study the amount of variation allowed from a target value in order to maintain quality through experimentation. The methods proposed by Genichi Taguchi have made this analysis popular. Taguchi's methods require fewer experiments to arrive at the desired information.

The purpose of this phase is identify items of breakthrough (levels not attained before) through the use of the dendrograms, then to determine the variance allowed to maintain quality through the use of factor analysis. (King, 1989:26-1 to 26-10)

❑ ***Chart F-4 The Design Improvement Plan.*** This chart identifies targets such as the cost, weight, etc, for both the current product's parts and the planned product's parts, and bottleneck engineering studies for the planned product's parts. In addition, the parts critical for function, sub-assembly, and reliability can also be listed. The result is a matrix which identifies new design goals as they relate to the original design. (King, 1989:27-1 to 27-5)

❑ ***Chart G-1 QA Table.*** As its name implies, this is a table rather than a chart. This table contains the parts, degree of importance, the part's relation to the A-4 chart, the quality characteristics, design specification, and problems if the design is not met. The purpose of this table is to identify critical design items and the potential problems associated with them, as well as improving communications between design and manufacturing. (King, 1989:28-1 to 28-4)

❑ ***Chart G-2 Equipment Deployment.*** This chart lists the various production parts that may be supplied from outside the company or in-house. The chart includes the various suppliers' (including in-house) quality and cost of the parts in

question. These are then compared and the supplier can be chosen. (King, 1989:29-1 to 29-4)

- ***Chart G-3 Process Planning Chart.*** This chart contains the process number, the process name, conditions of manufacturing including equipment and equipment settings, and control points in the process. The purpose is simply to identify how the process will be controlled. (King, 1989:30-1 to 30-4)

- ***Chart G-4 Process Fault Tree Analysis.*** This is similar to the other FTA type charts except that the process is analyzed instead of the product. The various ways in which the process can fail, and the causes of these failures, are identified to aid in determining the root cause of the failure. (King, 1989:31-1 to 31-4)

- ***Chart G-5 Process FMEA.*** This chart is also similar to the product FMEA chart. This chart lists the parts' processes, potential failure modes, potential effects of failures, and initial controls, as well as priority ranks the failure modes. A numerical representation of the likelihood of failure, seriousness of failure, and the probability of a product produced without detecting a failure. This chart provides a systematic way of identifying and controlling failure modes. (King, 1989:32-1 to 32-5)

- ***Chart G-6 QC Process Chart of Parts and Assembly.*** This table identifies the details of how the process will be controlled. The table lists the process number, process name, machines to be used, work instruction sheets, quality check items, control methods, and inspection methods. Additional information is included as needed to control the manufacturing process. (King, 1989:33-1 to 33-5)

These twenty seven additional charts, combined with the three charts developed in detail earlier, encompass the entire product development life cycle from customers'

requirements to production floor instructions. The reader can now see how the customers' requirements can be traced from the initial "Whats" expressed in the customers' own terms, to the manufacturing steps for the final product. We will now look at another QFD model which is characterized by a simpler four phase approach.

The Clausing/Makabe-ASI QFD Model. Each of the two models makes use of the QFD matrix. In the ASI model this matrix is referred to as the House of Quality (Eureka, 1988:18) as opposed to the Quality Chart (Akao, 1990:6). The two phrases may be used interchangeably.

In the ASI model only four basic charts (now referred to as matrices) are used. A specific matrix describes each of the four phases of the ASI QFD model. These four phases are (Eureka, 1988:37):

- Product Planning (Phase I).
- Parts Deployment (Phase II).
- Process Planning (Phase III).
- Production Planning (Phase IV).

Phases I and II can thought of as product planning and design, and Phases III and IV product process and production (Eureka, 1988:18). The results of each phase feeds into the beginning of the next phase. This is referred to as phase deployment and can be seen graphically in Figure 14. The rest of this chapter will discuss Phase I in detail and Phases II through IV briefly.

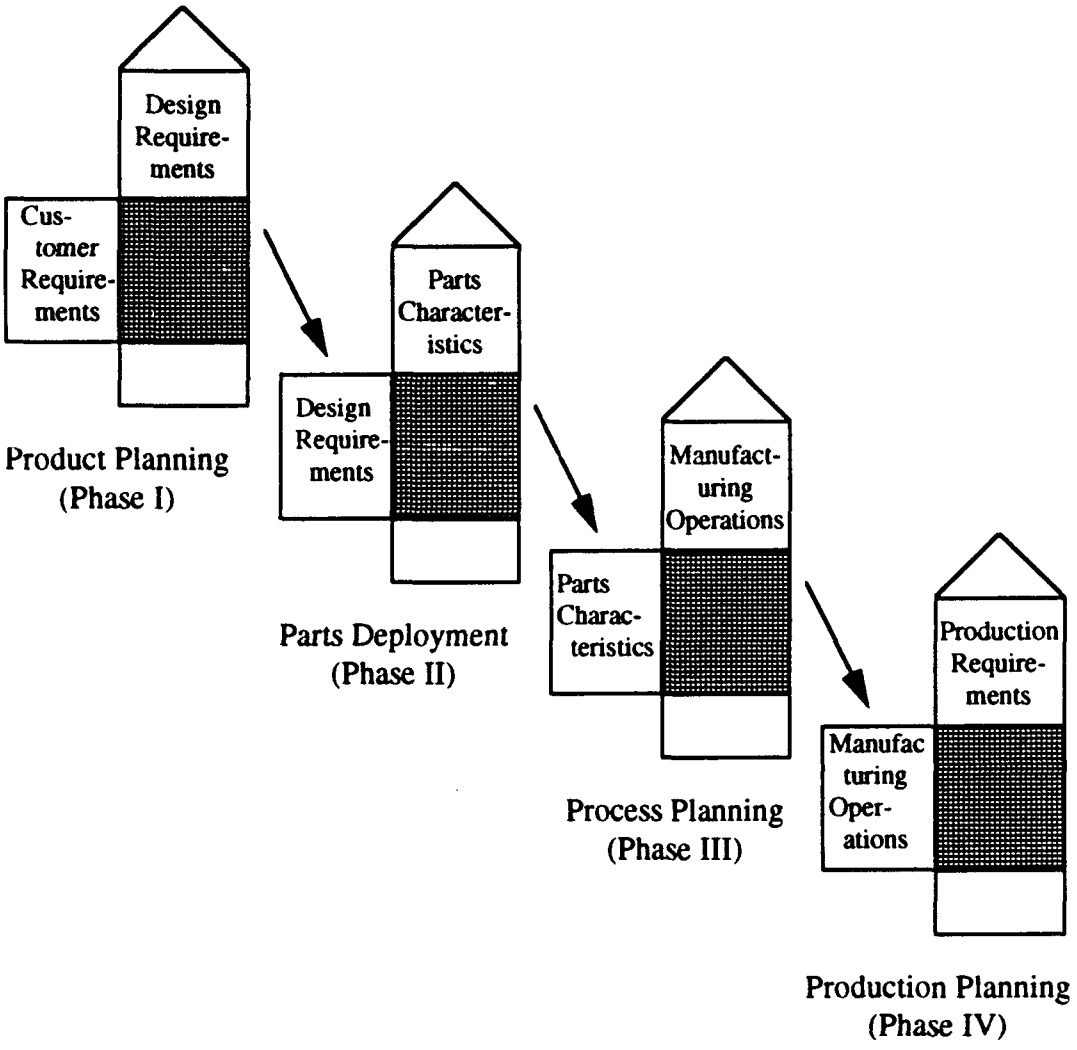


Figure 14. Phase Deployment (Eureka, 1988:37)

ASI Phase I Matrix. We will study the same sample problem of a mousetrap (Figure 15). The example matrix that follows is similar to the first three matrices to be prepared in the GOAL/QPC model (Akao, 1990:6).

As before, the matrix consists of a left and top side containing a hierarchy of items. These items are generally divided into primary, secondary, and tertiary levels of "Whats" (previously called customers' demands in the GOAL/QPC model) on the left and

corresponding "Hows" (previously called quality characteristics in the GOAL/QPC model) on the top. In the example, the basic "What", or customers' requirement, consists of the desire to have *no mice* in the house, to meet this we have the second level of customers' requirements *Easy to use*, *Effective*, and *Lure*. These secondary requirements are further broken down into subordinate requirements. As before, these customers' requirements may have been gathered from a survey or interviews with customers. They are vague by engineering standards, but strongly relate to the demands of the customer. (Eureka, 1988:18-22)

Across the top of the matrix are the "Hows." The primary level "How" is *killing* mice, broken down into its secondary and tertiary levels. The positioning of the "Whats" vice "Hows" form a matrix. Within this matrix are representative symbols that show relationships between the two.

The same symbols are used here as in the GOAL/QPC model. They are a triangle for a weak relationship, a circle for a medium relationship, and a double circle for a strong relationship, and no symbol for no relationship (Eureka, 1988:25-26; Akao, 1990:53). In our example, the desire for *foolproof* is weakly related to *size*, strongly related to the *dead mouse ratio*, and somewhat related to *time to kill*, *skid resistance*, *number of usable baits*, and the *setting force* required. There are no relationships to the other "Hows."

In developing a matrix such as this, the designers can see how the different aspects of the design of the mousetrap impact the basic customers' requirements and therefore its appeal to the customer. In addition, any rows or columns that do not contain a relationship, are areas that should be re-checked for deletion or change, providing a valuable cross-check (Eureka, 1988:26).

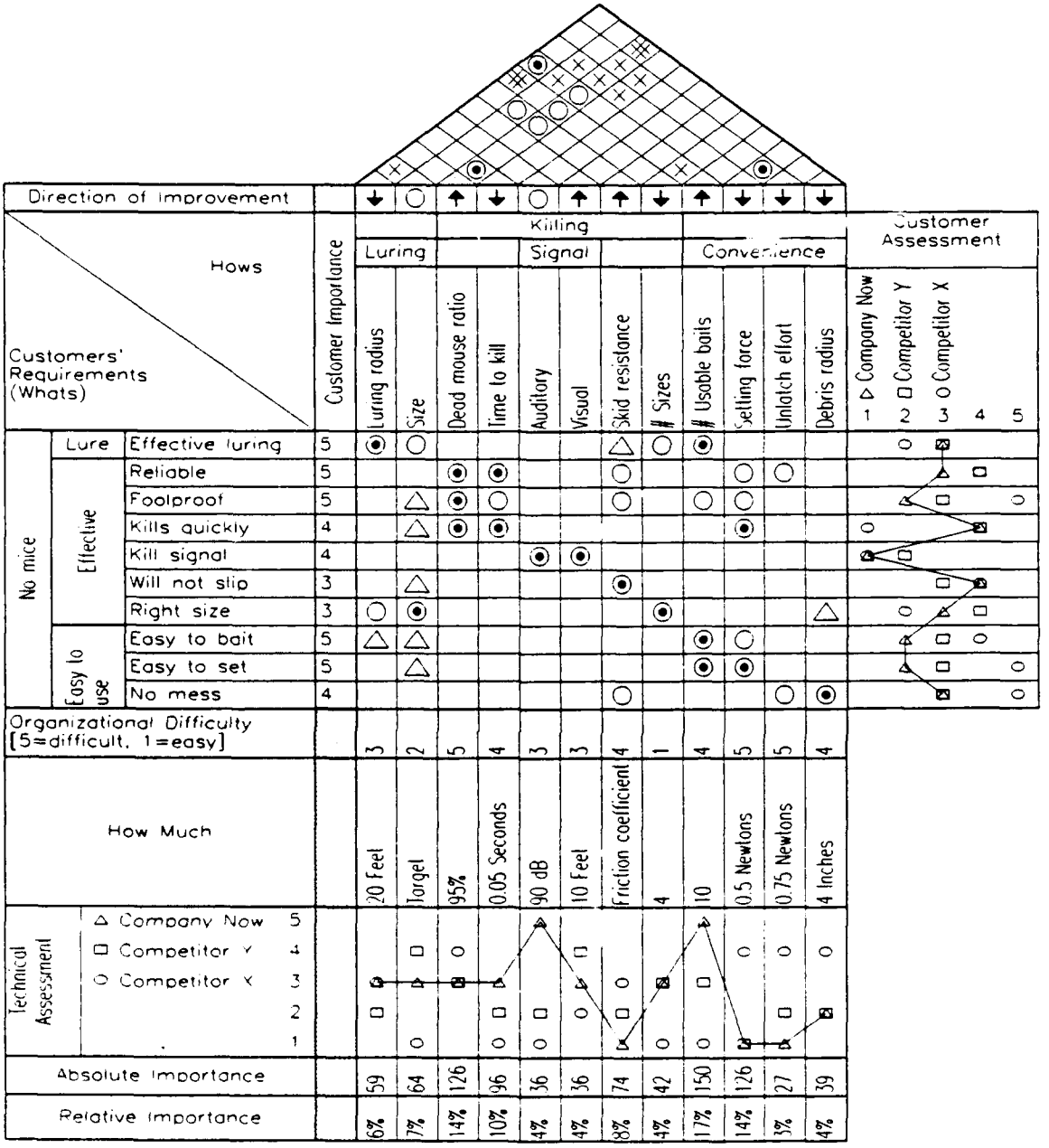


Figure 15. ASI Phase I QFD Matrix (QualiSoft, 1991)

After the "Hows" have been decided, a row is added to the bottom of the matrix which reflects the "How Much" best expressed in terms of a numeric goal (Eureka, 1988:28). In our example, 95% of the mice trapped should be killed, the *setting force* of

the trap should be 0.5 newtons, and the *time to kill* the mouse should be 0.05 seconds. Each attribute can be tested or measured in some way to show achievement of the goal. Each attribute can also be associated with a desired direction for improvement. This is represented by the arrows as in the GOAL/QPC model. In our example we would like to kill the mouse in less than 0.05 seconds as opposed to greater than 0.05 seconds.

Once the "Whats" versus "Hows" relationships have been determined, the "Hows" versus "Hows" correlation is determined. This is referred to as the correlation matrix and forms the characteristic roof on the top of the house of quality. As in the GOAL/QPC chart A-3, this matrix can show how the various "How" items support each other. The correlation is shown using a double circle for a strong positive correlation, a single circle for some positive correlation, a cross showing some negative correlation, and a double cross showing strong negative correlation. (Eureka, 1988:31) In our example, *dead mouse ratio* has a strong positive correlation to *time to kill*, while *dead mouse ratio* has a strong negative correlation to *debris radius*.

The next step establishes a competitive assessment for each item of both the "Whats" and "Hows." The assessment of the "Whats" is usually called a customer competitive assessment and the assessment of the "Hows" is usually called a technical competitive assessment. These technical assessments generally use engineering generated data. A check can be performed here as a form of sanity check. Strongly related "Whats" and "Hows" should exhibit similar customer and technical assessments. Those that do not may indicate an error in engineering judgement. As in the GOAL/QPC chart A-1, the assessments may be given a numeric value of between one and five. The customers importance can then be weighted by the relationship (nine for strong, three for medium, and one for weak) and used to form a new horizontal row showing absolute importance. This can identify critical areas which may benefit from additional effort.

Some additional elements can also be added such as selling points, level of technical difficultly, technical standards, and quality standards. (Eureka, 1988:33-35)

The ASI Phase I matrix presented earlier shows most of the elements described above. Most of the information presented in this one matrix is equivalent (or can be made equivalent) to the information presented in the first three charts of the GOAL/QPC model.

We have studied in detail the first phase of the ASI model. The subsequent phases will be briefly discussed but are provided only so the reader can conceptualize the entire ASI QFD model. Phases II through IV are not within the scope of this research effort.

ASI Phase II Matrix. This phase is called the parts deployment phase. The key design requirements from the previous phase are brought down into this matrix. Those design requirements that are already being achieved are not brought down to this matrix to avoid wasting time and effort. This is a matrix similar to the Phase I matrix but describes the requirements in more precise engineering terms. Activities such as value engineering, fault tree analysis, failure mode and effects analysis, cost analysis, and parts selection for reliability can also be accomplished in this phase. The result of this phase is the identification of the critical part characteristics necessary to meet the design requirements. (Eureka, 1988:36-38)

ASI Phase III Matrix. This phase is called the process planning phase and follows the transition from design to manufacturing process planning. A process planning chart is prepared for each critical part characteristic as identified from the previous matrix. This process planning chart includes a listing of the required processes, a matrix relationship between each process and each critical part characteristic,

and a listing of process control parameters. A process failure mode effects analysis may be conducted and the previous matrix is verified. The result of this phase is a process control chart for each part. (Eureka, 1988:38-39)

ASI Phase IV Matrix. This phase is called the production planning phase and transitions from the previous phase down to the production floor. This takes the form of various tables and charts tailored to meet the specific needs of the user. (Eureka, 1988:39-40)

Software Quality Function Deployment

We will refer to the application of QFD to software development as Software Quality Function Deployment (SQFD). A few companies have begun to use SQFD in various degrees. These companies include AT&T, Hewlett Packard (HP), Digital Equipment Corporation (DEC), International Business Machines (IBM) CSK, Nippon Systems, and others (Akao, 1990:331; Brown, 1991). Companies are not quick to share detailed information on their use of SQFD or products being developed with SQFD. This is partly because SQFD is being used on forthcoming products and therefore is considered sensitive (Brown, 1991).

One researcher/private consultant has made SQFD more available to the public through the publication of a paper on the subject. Mr Richard Zultner is considered to be an expert in SQFD and has developed an SQFD model and also created a course in SQFD (GOAL/QPC, undated:22). His approach to integrating QFD and software development, together with some information obtained from IBM, will be presented as the current literature on the subject.

Software development is a process oriented industry as opposed to the product oriented manufacturing industry. The QFD methodology works well with the ideas of building a product. The "Whats" and "Hows" are things that the customer can see and feel, the engineer can quantify and design. In the process industry the analogue is not as clear. In software development, the output of one process feeds the input of the next process, thereby amplifying the effects of errors in the previous process. This must then be remedied in the later processes at greater expense and time and sacrificing efficiency (Conti, 1989:46). QFD holds promise as a tool to focus the chain of processes so that the final output meets the customer's requirements (Conti, 1989:47).

SQFD has been used by limited numbers of software developers beginning in 1982 (Akao, 1990:331). The small numbers of companies involved in SQFD has made it difficult to refine an SQFD process (Akao, 1990:331). We will outline an SQFD process based upon the process proposed by Zultner, with some slight modifications, and how it compares to the hardware QFD methods discussed earlier.

A Possible SQFD Model. The Zultner model follows the GOAL/QPC approach using fourteen possible matrices. Three Z-series matrices are added that integrate with the A-series matrices of the GOAL/QPC model. D-series and E-series matrices are also represented. We will only examine the Z and A-series matrices used as part of this research effort. The remaining Z, A, D and E-series matrices are beyond the scope of this research effort and will only briefly be described. (Zultner, 1990:135)

Z-Series SQFD Matrices. The initial matrix, the Z-0 matrix, seeks to identify the potential stakeholders and their interests in the product. An example of this matrix is included in Chapter IV. This matrix does not show organizational relationships but relationships involving the roles of different customers in the project.

These relationships can be financial, policy making, frequent users, etc. The matrix has users (the terms user and customer are considered identical) or stakeholders across the top, and user characteristics (interests) down the left hand side. Relationships will be identified using the standard three symbol set (and values) as opposed to the five symbol set proposed by Zultner. An importance column is also included down the left hand side, rating the importance of the various user characteristics. When this matrix is solved, the resulting values indicate the relative importance of the users, or stakeholders, in the system. These values of importance are carried through to the Z-1 matrix. (Zultner, 1990:134)

The Z-1 matrix attempts to determine the users' requirements. An example of the Z-1 matrix can be found in Chapter IV. This matrix brings forward the users from the Z-0 matrix and their relative importance. The users are arranged across the top of the matrix with their relative importance across the bottom. The users' requirements are gathered and listed in hierarchical format down the left hand side. An importance column for the users' requirements is also included. These requirements are not the technical requirements that will be determined later, they are instead the users' expectations of what the system will do. Relationships are determined and represented using the standard three symbol set. The relative importance of the users is multiplied by the relation and summed across the rows. This raw priority is then multiplied by the importance of the users' requirements and represents the adjusted priority of the users' requirements. This priority takes into account the relative importance of the various users determined in the Z-0 matrix. This adjusted priority is brought forward to the next matrix in the process, the A-1 matrix. (Zultner, 1990:134-135)

A-Series SQFD Matrices. The next matrix to be constructed is the A-1 matrix. An example of the A-1 matrix can be found in Chapter IV. This matrix

results in a priority of the users' technical requirements. The users' requirements are brought forward from the Z-1 matrix along with the adjusted priority values. Across the top of the A-1 matrix are the technical requirements necessary to meet the users' requirements. These technical requirements should be in terms of what processes need to be done and be measurable in some way. The users' requirements may need to be adjusted based on the rate of improvement, sales point, or possibly competitors' positions. Relationships between users' requirements and technical requirements are determined and represented using the standard three symbol set. The requirements weight, a percentage determined by multiplying the adjusted priority, rate of improvement, and sales point, is calculated. This requirements weight is then multiplied by the relationship value and summed down the column of technical requirements. The resulting technical priority of each technical requirement represents the importance to the user of that particular requirement. Additional information, such as units of measure, target values, and direction of improvement can also be added. The technical requirements and their priorities are brought forward into the A-2 matrix. (Zultner, 1990:135-136; Sharkey, 1990:Charts 10-12)

The last matrix to be studied in detail here is the A-2 matrix. An example of this matrix can be found in Chapter IV. The A-2 matrix, as presented by Zultner, shows relationships between the technical requirements and the processes and entities required to perform them. These processes and entities are determined through the structured analysis techniques (or other methods) and are listed down the left hand side of the matrix. We will only study the processes and not the entities and shift the processes from the left of the matrix to across the top. The technical requirements and technical priorities are brought forward from the A-1 matrix and listed down the left hand side. The processes as determined by structured analysis (based on the information gathered so

far) are arranged across the top of the matrix. Relationships are again determined and represented using the standard three symbol set. The matrix is solved by multiplying the technical priorities by the relationship value and summed down the column of processes. The resulting priority represents the value to the user of that particular process in the system. (Zultner, 1990:136-137; Sharkey, 1990:Chart 16)

Additional SQFD Matrices. Several additional SQFD matrices exist but were not used directly in this research effort (Table 3). They are important however in understanding how SQFD relates to QFD. The table below shows the SQFD matrices along with their QFD counterparts, and the attributes they analyze.

Table 3
Comparison of SQFD and QFD Matrices

SQFD Matrix	Attributes	QFD Matrix	Attributes
Z-0	Users vs. User Characteristics	N/A	N/A
Z-1	Users vs. User Requirements	N/A	N/A
Z-2	Entity vs. Process	N/A	N/A
A-1	Technical Requirements vs. User Requirements	A-1	Quality Characteristics vs. Customer Demands
A-2	Technical Requirements vs. Entity/Process	A-2	Quality Characteristics vs. Functions
A-3	Technical Requirements vs. Technical Requirements	A-3	Quality Characteristics vs. Quality Characteristics
D-0	Failure Modes vs. New Technology	N/A	N/A

D-1	Failure Modes vs. User Requirements	D-1	Failure Modes vs. Customer Demands
D-2	Failure Modes vs. Entity/Process	D-2	Failure Modes vs. Functions
D-3	Failure Modes vs. Technical Requirements	D-3	Failure Modes vs. Quality Characteristics
E-0	New Concepts vs. New Technologies	N/A	N/A
E-1	New Concepts vs. User Requirements	E-1	New Concepts vs. Customer Demands
E-2	New Concepts vs. Entity/Process	E-2	New Concepts vs. Functions
E-3	New Concepts vs. Technical Requirements	E-3	New Concepts vs. Quality Characteristics

Some of the SQFD matrices do not have QFD counterparts. Those that do represent the analogous software attributes for the different matrices. Additional work is still needed to determine analogs for other SQFD matrices from their QFD counterparts (Zultner, 1990:141).

SQFD Experience. A few companies have published general information of their use of SQFD. We will briefly review the use of SQFD at AT&T and Hewlett Packard.

SQFD at AT&T. AT&T began their study of SQFD in 1986. They have since applied the first phases of SQFD to several projects with very positive results. AT&T finds that SQFD helps to translate the customers' needs into technical terms, and provides traceability of those technical terms back to the customers' needs. SQFD also helps to focus more specifically on the customers' real needs, aids in communication among other AT&T organizations, and provides a means of documenting the history of

the analysis process. SQFD fits well into the overall philosophy at AT&T of constantly providing better products that best incorporate the customers' needs. (Thompson, 1989:279-280)

AT&T cites three projects that have used SQFD. The AT&T SQFD model includes five basic steps:

1. Identify market needs
2. Prioritize needs
3. Identify technical characteristics and features
4. Map technical characteristics and features to needs
5. Prioritize technical characteristics and needs (Thompson, 1989:281)

SQFD however, was specifically tailored to the needs of each project. The use of SQFD resulted in better understanding of the customers' needs, quicker product definition (17% time reduction), improved communications and teamwork, and a view of the system from the customers' perspective. (Thompson, 1989:285)

The general benefits from SQFD are seen as an increase in their customers' satisfaction, less time spent on development of products, and fewer changes in requirements. AT&T will continue to develop their SQFD process and apply it to more projects in the future. (Thompson, 1989:279, 285)

SQFD at Hewlett Packard. SQFD was first introduced at HP in 1986. HP formed the Industrial Application Center (IAC) in 1986 with the mission to improve quality, responsiveness, and productivity. The IAC began work on a project called Interactive Visual Interface (IVI) and this was used as a pilot program to introduce the use of SQFD at HP. The first step in implementing SQFD was to sell the idea to top management. This was accomplished through a presentation. The second step was to identify the pilot program. Lastly, the program personnel, including higher management,

were trained in the use of SQFD. A consultant was also hired as an SQFD expert to help the IVI team. (Shaikh, 1989:290-293)

Customers' needs were collected on 3 x 5 post-it notes during customer visits. These needs were organized into a three level hierarchy using affinity diagrams (one of the seven Japanese management tools) by the project team. A tree diagram (another of the seven management tools) was used to fill in any gaps in the customers' needs. Importance ratings were identified by the IVI team for each of the tertiary customers' needs. Next, the methods needed to satisfy the customers' needs were brainstormed, and using affinity diagrams, arranged into a three level hierarchy. Again, a tree diagram was used to help fill in any gaps. Relationships were established by mini-teams of experts and novices. This team make up was meant to balance the expert knowledge with the novice's questioning, forcing a re-evaluation of the relationships. This process was later changed to an individual task to expedite the effort. (Shaikh, 1989:294-295)

The resulting weighted priority was used to determine which features were to be incorporated into the first release of the product. Features that were rated a five, and some that were rated a four were planned to be included in the first release. Competitive assessments were conducted of similar software products to determine where customers' needs were already being met, and where they were not. Target values were also identified. (Shaikh, 1989:295-296)

Defect data was collected on similar projects in an attempt to determine why these defects occurred, hoping to avoid these defects in the IVI project. Control items, such as government mandates, marketing strategy, HP's current market position, were identified for certain customers' needs (demands). Metrics were also designed to help

track the effectiveness of SQFD. These measured engineering hours and the number of features added or removed from the product. (Shaikh, 1989:296-297)

Some of the benefits cited include: considering the voice of the customer throughout the life cycle, determination of features based on the customers' needs, prioritizing engineering resources with the most important customers' needs, its benefits as a communications tool among project personnel, and competitive analysis useful in determining strengths and weakness of the product. Lessons learned include: beginning SQFD in the development phase of a project, get the proper training and guidance on the use of SQFD before starting, and having a software tool to keep the SQFD matrices up to date. In general, HP found SQFD to be a valuable tool but only one of many tools to be used during product development. Pilot projects require an investment of time, resources, and training and must have top management's baking to be successful. (Shaikh, 1989:298-299)

Summary

This chapter began with a review of the structured analysis techniques including the use of data flow diagrams, data dictionaries, and entity-relationship diagrams. Some shortcomings of these techniques identified were the need to better incorporate the users' real requirements into the structured analysis, a lack of a means to prioritize the users' requirements, and a lack of a means to trace the users' requirements from the users' voice into the structured analysis representation. Next, we reviewed two QFD models, including an example problem of a mousetrap. Using QFD we saw how the customers' needs (demands) were translated into the design requirements of the mousetrap, and how follow-on QFD matrices make further use of these needs. We saw how QFD could be

applied to software development and outlined one such approach. Finally, a few case studies of actual SQFD experiences were reviewed.

III. Methodology

Introduction

This chapter outlines the methodology that was used to determine if SQFD may be used by the USAF to aid in requirements analysis and definition and/or should be further investigated. The results are presented in Chapter V.

Validation Approach

Each investigative question was answered with a methodology appropriate for the type of question. The two types of validation that were used were a review of the current literature, and a survey of knowledgeable personnel. The survey will be discussed in detail later in this chapter. The validation approach for each investigative question is presented below.

Investigative Questions. A review of the six investigative questions and their purposes is listed below. The methods used to validate these questions is contained in the next two sections.

- IQ1 - What are the QFD process, its goals, products and techniques?

The purpose of this question was to determine the present QFD process in order to better understand how the process may be tailored to the software development model. This understanding formed the basis of the software specific QFD methods that would be used in Chapter IV.

- IQ2 - What are the current techniques available to develop software systems requirements for the USAF?

The purpose of this question was to determine present methods and techniques of requirements analysis and their shortcomings. It is these shortcomings that the tailored approach to QFD is intended to help overcome.

- IQ3 - Can the QFD process be tailored to meet a specific domain's requirements?

The purpose of this question was to determine if the QFD process can be tailored to meet specific requirements, or if the QFD process is domain specific, e.g., to the automotive industry. Without this knowledge it would have been futile to proceed.

- IQ4 - What specific tailoring of the QFD process should be made to use QFD to aid in developing software systems requirements in the USAF environment?

The purpose of this question was to determine the specific methods to use in applying QFD in the software development domain. This knowledge was key to the research effort. These specific methods would later be used to develop a sample problem which will be compared to a structured analysis representation of the same problem.

- IQ5 - Is the QFD process acceptable to USAF software managers/ engineers?

The purpose of this question was to determine if the QFD process can be used in the USAF environment. If the QFD process does show promise of providing benefits to the area of software requirements analysis, can the process fit into the USAF way of doing business.

- IQ6 - Does the application of the tailored QFD process result in requirements analyses that correct shortcomings of current techniques?

The purpose of this question was to determine if the application of QFD can overcome some of the shortcomings of the current techniques in requirements analysis as identified above.

Validation for Investigative Questions 1 - 4. The first three investigative questions and part of the forth investigative question were satisfied through a review of the current literature. It was intended that the literature presented in Chapter II would be adequate to answer the first three investigative questions. This literature was gathered through professional journals, instructional material, and books on the subjects, and represents current knowledge on the topics.

The fourth investigative question involved two types of knowledge. The first type was that gained from studying currently used techniques. The second type was that gained through an applied problem. The literature presented in Chapter II and the methods used in Chapter IV were intended to satisfy this investigative question from a standpoint of current knowledge. This included a review of previous attempts at tailoring the QFD process for software applications and any previous successes in this area.

Validation for Investigative Questions 4 - 6. Investigative questions five and six and part of investigative question four were to be answered by examination of the applied problem. A sample problem was presented to various knowledgeable software personnel through the use of a survey. The responses obtained from these personnel via the survey would satisfy these investigative questions. This survey formed the basis of this research effort and will now be discused in greater detail.

Survey Methodology

The best way to validate a new requirements analysis technique would be to develop multiple identical systems each using different requirements analysis methods. The systems could then be directly compared to one another on the basis of pre-determined criteria in order to judge which was the better requirements analysis method. This form of validation would demand more time and effort than that available for this research effort. In addition, before such a methodical validation approach is undertaken, the basic question of feasibility must be shown. Until the feasibility of the new method is confirmed, the time and resources for a more robust method cannot be justified. This basic feasibility was the goal of this study.

The validation approach used for this research effort must be within the scope of time and resources available so that it could be completed and results determined. Therefore, the approach used utilized the opinions of knowledgeable personnel on the application of the software specific QFD techniques as demonstrated in the survey package.

The basic idea behind the survey was to identify and recruit knowledgeable people in the field of software engineering and have them review a package of information (see Appendix A) and provide their opinion. This package of information was produced using the specific SQFD techniques identified in Chapter II. These techniques showed how SQFD fits into the overall software development model, how it might be used to aid in the requirements analysis phase of the model, and how SQFD is integrated with current requirements analysis techniques.

For this research effort, the knowledgeable personnel came from the areas of software acquisition management, software engineering, user organizations, and the

academic arena. A minimum of ten responses were sought with fourteen actually being received. The opinions were gathered through responses to questions posed about the information package.

As applied to this research, the survey:

- **Presented a sample software requirements problem**

The sample problem was based on the TELLERFAST software system which, together with the automatic teller machine (ATM) formed the AUTOTELLER Automatic Teller System (Loy, 1990:441). The TELLERFAST software performs all of the functions necessary for the AUTOTELLER system to operate (Loy, 1990:443). These functions included:

- Accepting and validating the ATM cards.
- Supplying and responding to user menus.
- Issuing cash.
- Accepting deposits and loan payments.
- Transferring funds among accounts. (Loy, 1990:443)

This problem was chosen for the following reasons:

- It was a relatively simple problem and could be easily understood by those who were asked to provide a survey response.
- It was developed by an independent source and therefore reduced bias.

❑❑ It presented a coherent problem so that a comparison of requirements analysis techniques could be made.

❑ **Described the problem using two requirements analysis methods, a baseline method and the new method**

The sample problem was described using the structured analysis method as a baseline. This analysis has been accomplished in the TELLERFAST specification and was used, with some simplification, for the survey package.

An SQFD representation of the same sample problem was also developed. This new representation followed the same objectives and constraints as the baseline representation of the TELLERFAST system.

Both the SQFD representation and the traditional structured analysis representation were included in the survey package.

The SQFD representation was prepared by the use of an automated tool. The tool used was called QFD Designer version 2.0 developed by QualiSoft Corporation. The tool can be obtained directly through QualiSoft or the ASI. This particular tool was chosen over the leading competitor (QFD Capture) because of its flexibility and ease of use. This automated tool was integrated into the specific techniques presented in Chapter II. Slight modifications were made to the SQFD method to better utilize the tool and accomplish the SQFD matrices. This was a valid constraint as the QFD process is unmanageable and unwieldy for large problems if attempted by hand (Porter, 1989:323). The automated tool both aided in the use of the SQFD process and could ease its acceptance as a viable technique.

- **Gathered knowledgeable persons' opinions by reviewing the two requirements analysis methods**

The knowledgeable personnel were provided with the package of information which contained a brief explanation of the SQFD process and the two representations of the TELLERFAST problem. They were then asked nine subjective questions in order to gather the information necessary to answer the investigative questions.

These questions were:

1. Do you think that the structured analysis representation of the sample problem (not taking into account the SQFD analysis) adequately captures all the users' wants, needs, and desires? Please explain.

2. Does the SQFD approach help to capture any of those needs, wants, desires, you felt were missing (if any) from the Structured Analysis approach? Please explain.

3. Key to the SQFD process is capturing the importance (or priority) of the various characteristics of the problem. Do you believe that this is missing from most Structured Analysis representations? Is this useful information to determine?

4. In your opinion, would the use of SQFD provide a better means of requirements traceability then methods you currently employ? Please explain

5. The SQFD Z-0 matrix attempts to analyze the different users and their relative importance or clout. From your experience, do you believe that this would be useful information? Please explain.

6. Do you believe that the additional information resulting from using SQFD could result in a better Structured Analysis of the problem? In other words, would SQFD

provide a better front end to Structured Analysis then present analysis methods? Please explain.

7. Based on your experience, are there any reasons why SQFD would not be "workable" in the USAF environment? Are there obstructions that would not make SQFD possible to use?

8. If you were trained to use SQFD, would you seek to apply it to a real-world program?

9. Any additional comments you may have would be greatly appreciated, including any suggested modifications to the SQFD process.

The survey responses were then analyzed to determine if there was a consensus and to draw conclusions on SQFD's viability in the USAF software arena. The complete information package is provided as Appendix A.

Administration of the Survey. The knowledgeable personnel were determined through personal knowledge of the researcher and further recommendations from those knowledgeable people themselves. A total of twenty two surveys were sent out. The distribution of the surveys were as follows:

- four to Academia
- four to software acquisition managers
- ten to software engineers
- and four to user organizations

Follow up calls were placed to those failing to return the survey by the deadline. A total of fourteen survey responses were received. The raw survey responses are included in Appendix B for the reader's own review.

The analysis of the survey responses and the answers to the investigative questions can be found in Chapter V.

IV. Application of Software Quality Function Deployment

Introduction

This chapter presents two representations of a sample problem. The sample problem is described below. The scenario will be used to develop a structured analysis representation using a context diagram, data flow diagrams (DFDs), and a data dictionary. The same scenario will also be used to develop a software quality function deployment (SQFD) aided representation. This representation will utilize some of the structured analysis information as well.

Sample Problem Scenario

The sample problem will be based on the TELLERFAST software system which, together with the automatic teller machine (ATM) forms the AUTOTELLER Automatic Teller System. This new system is intended to replace the existing teller machines. The TELLERFAST software performs all of the functions necessary for the AUTOTELLER system to operate. These functions include:

- Accepting and validating the ATM cards.
- Supplying and responding to user menus.
- Issuing cash.
- Accepting deposits and loan payments.
- Transferring funds among accounts.

There are two categories of users of this system. Bank customers are those users that interact with the system to conduct a transaction. Bank employees are those users

that interact with the system to perform maintenance, query the system, and recharge the teller machines with supplies, including money. The next two sections describe the two users' requirements in narrative form. The requirements were "reverse engineered" from the AUTOTELLER software requirements specification and simplified for use here (Loy, 1990:439-456). The full specification can be found in Appendix C. These requirements will be used throughout this chapter.

Customer Oriented Requirements. A survey was conducted of 250 regular users of automatic teller machines (ATMs). The goal of this survey was to better understand the customers' use and expectations of ATMs in order to produce a better system. The users were asked to rate the importance of each feature on a scale of one to five with five being the highest. The survey resulted in the following findings:

- A majority of the customers liked a menu system to prompt them through an ATM transaction. This feature was given a five.

- Most customers appreciated the use of audible signals to remind them to remove their money, receipt, and ATM card. This feature was also rated a five.

- The minimum types of transactions the customers expected with their importance ratings were as follows.

    - Balance inquiry - three
    - Deposit checks - two
    - Transfer funds between accounts - three
    - Loan payments - one
    - Withdrawal of cash - four

- An informative receipt should be issued at the end of each transaction. The receipt should have, as a minimum, the date, the account number, the amount of the transaction, and the finishing account balance. This feature was assigned a value of four.

In addition to the survey, fifteen volunteer customers were used to conduct a assessment of the response times desired for the ATM operation. The customers used a specially modified ATM under the supervision of the ATM contractor. The results of this study were:

- The response time for menu changes should be no more than three seconds.

- The response time to read an ATM card should be no more than three seconds.

- Sixty seconds should be allowed for envelope insertion when a deposit is requested. For all other prompts, the customer should have thirty seconds to respond.

- The reccipt should be issued no longer than two seconds after the session is terminated.

- Response times in general were felt to be important and were rated a five.

This is the entirety of the customers requirements.

Bank Employee Oriented Requirements. Requirements from the bank employees were gathered through face-to-face discussions with employees of the bank including tellers, bank managers, and ATM system maintenance personnel. The following list comprises all the bank employees' requirements. As before, each feature's importance was rated on a one to five scale.

- The system is required to interact with the bank's existing central computer system.

- The system must verify the customer's ATM card and Personal Identification Number (PIN) are correct. Three chances will be given to correct the PIN. A list of stolen and lost cards will be maintained. Each ATM card number will be checked against this list for a match. Any lost or stolen cards will be kept by the machine and the customer informed. This feature was given a value of five.

- If an ATM machine cannot dispense cash due to insufficient funds in the machine, the session should be terminated with an error message and the machine put on stand-by. This was also given a four.

- The system will maintain information on the last withdrawal and will not allow more than $200 to be distributed in any given day. An error message will be issued in these instances. This feature is only rated a two.

- The system will perform the following accounting functions as a minimum:

    - Maintain account balances of the customer's accounts. Very important and rated a five.

    - Maintain the customer's deposit limit, credit card limit, daily withdrawal limit, and account withdrawal limit. These limits will be changed to reflect the customer's withdrawals. Not as important and rated a two.

- The following statistics and status information will be maintained.

- Number of customer sessions - two.
- Number of transactions completed - two.
- Number of each transaction type completed - two.
- Current amount of cash left - four.
- Number of customer receipts remaining - two.
- Amount of cash dispensed - four.
- Dollar amount expected in the depository vault - four.

- Computational errors must never occur, and the system must never lock up due to customer errors. This was considered very important and was rated a five.

- Adequate security safeguards will be built into the system. This is also considered very important and rated a five.

- The system can be out of service no more than 0.001% of its yearly operating time. Regularly scheduled maintenance is not included. This is also rated a five.

Structured Analysis Representation of the Sample Problem

Both of the above sets of requirements (from the bank customers and from the bank employees) were considered to be requirements of the AUTOTELLER system. All these requirements were used to arrive at the structured analysis representation that follows.

AUTOTELLER Context Diagram. The first step in decomposing the system is to construct a context diagram. A context diagram is intended to form the

overall concept of the system. It is a precursor to constructing lower level data flow diagrams and can be considered a type of domain analysis tool. After careful analysis of the users requirements presented above, the context diagram (Figure 16 below) was developed. The context diagram for the AUTOTELLER system shows four external entities or terminators and the central transform which represents the ATM system.

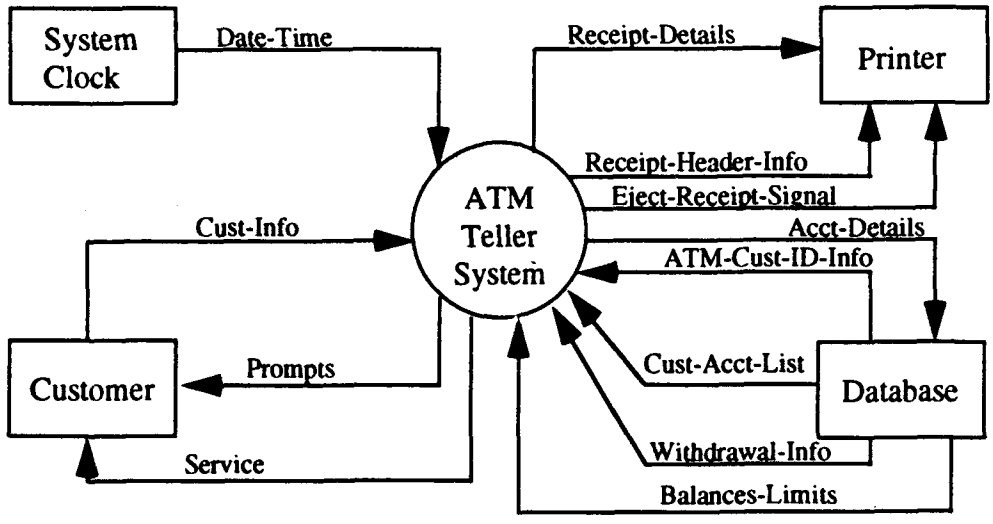


Figure 16. AUTOTELLER Context Diagram (Loy, 1990:454)

Each external entity was felt necessary to properly separate the system into logical components. The data flows between these components make up the total of all information needed at this level to satisfy the various needs of the customers.

- The *System Clock* supplies the time and date information to keep track and record the time sensitive information required for certain operations.

- The *Customer* provides the system with information such as his/her PIN, a requested transaction, and the transaction amount. The customer is supplied with

prompts from the system requesting action of the customer and service which takes the form of cash dispensed/deposited, loan payments, and account balance information.

- The *Printer* prints and dispenses a receipt for a transaction. The printer is supplied with header information (date, time, location, and customer ID), receipt details of the transaction to be printed, and an eject signal.

- The *Database* contains all the information that the systems uses to accomplish a transaction. This includes account balance information, withdrawal limit information, customer information, and is updated with current information (such as after a transaction is completed).

- The *ATM Teller System* accomplishes all processing necessary for the AUTOTELLER system.

Data Flow Diagrams. The central transform of the context diagram does not contain sufficient information to design the AUTOTELLER system. The central transform "bubble" is therefore expanded to include more details of its internal processing. This top level data flow diagram is shown in Figure 17 below. All four external entities and their data flows in and out are the same. However, the central transform has been expanded into four numbered processes: *Verify Customer ID*, *Initiate Transaction*, *Process Transaction*, and *Perform Accounting*.

This top level DFD helps to better define the system, but still more detail is needed to adequately define the system for a preliminary design. The top level DFD will therefore be further "leveled" to a second level set of DFDs. The first process is simple enough that no DFD is needed to describe its processing. However, the other three processes will require individual DFDs. A much more detailed understanding of how the

AUTOTELLER system will meet the customer's needs can be gained through constructing and reviewing these lower level DFDs. Each process is described in the next four sections.

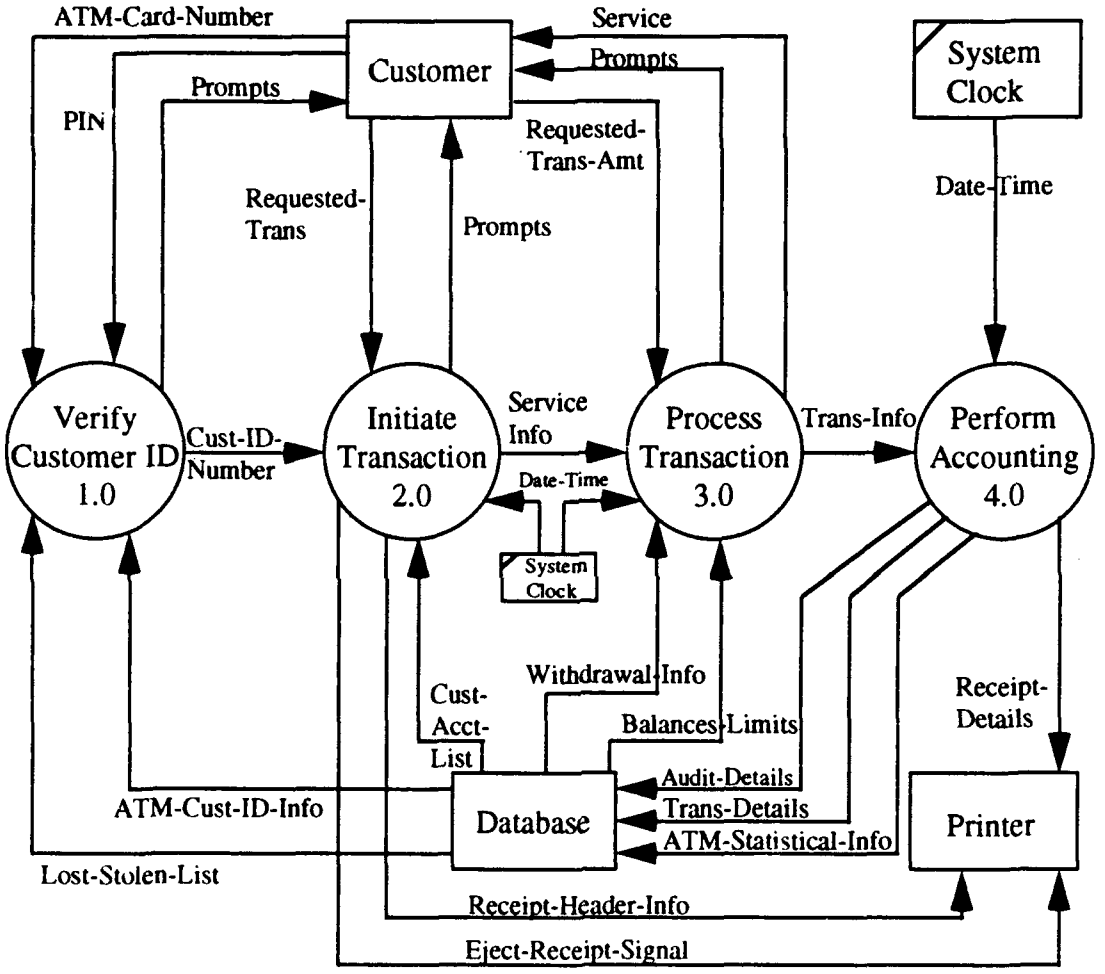


Figure 17. Top Level DFD (Loy, 1990:455)

1.0 - Verify Customer ID. This process receives the card number and PIN from the customer as well as lost/stolen card information and customer information from the database. It is responsible for comparing the card number to the lost/stolen list and to retain the card if a match is found. This process must also prompt

the customer for his/her PIN and verify the PIN against the database for correctness. The process passes the customer ID number to the Initiate Transaction process.

2.0 - Initiate Transaction. As can be seen from the DFD (Figure 18), this process had to be further broken down into three subprocesses.

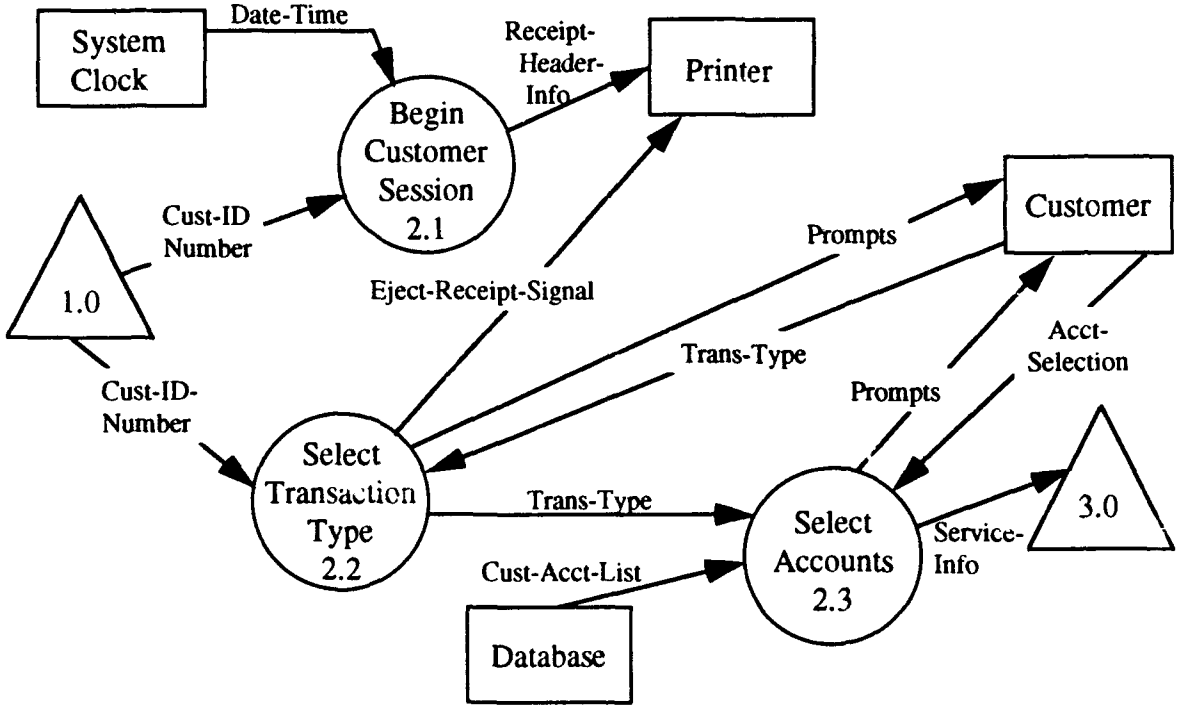


Figure 18. Process 2.0 - Initiate Transaction DFD (Loy, 1990:455)

Subprocess 2.1 - Begin Customer Session receives the customer ID number from Process 1.0 (as represented by the numbered triangle) and the date/time from the system clock. This process outputs data to be printed at the top of the customers receipt to the printer.

Subprocess 2.2 - Select Transaction Type receives the customer ID from Process 1.0 and the transaction type from the customer. This subprocess prompts the customer

for the transaction type and allows four transactions to be processed during one session. If the transaction chosen is *not terminate*, then the transaction type is sent to Subprocess 2.3. When four processes have been performed or the customer terminates the session, the eject receipt signal is sent to the printer to eject the customer's receipt.

Subprocess 2.3 - Select Accounts receives the account selection from the customer, the transaction type from Subprocess 2.2, and the customer account list from the database. It uses this information and sends prompts to the customer to determine which account the customer wishes to affect. Depending on the transaction type there may be one or two accounts affected. The process outputs service information to Process 3.0.

3.0 - Process Transaction. This process is the primary process of the top level DFD. It also required further definition, and was broken down into four subprocesses (Figure 19).

Subprocess 3.1 - Issue Cash receives a valid amount from Subprocess 3.3 and checks the supply of funds in the machine. If sufficient cash is available the service amount is sent to the cash supply for dispensing to the customer. If sufficient cash is not available an error message is displayed and the machine is put on standby.

Subprocess 3.2 - Display Balance receives service information from Process 2.0 and the current balance from the database. The account designation and its balance are displayed to the customer.

Subprocess 3.3 - Compare Amount Requested with Limits receives withdrawal information, balance and account limits from the database, service information from Process 2.0, transaction amounts from the customer, and the date and time from the

system clock. This subprocess checks to see if the customer has withdrawn funds already this day, and if the funds withdrawn earlier this day exceed $200. If not, the customer is allowed to withdraw funds that do not exceed the customer's limits. The customer is prompted for the transaction amount. After processing, the valid service information is sent to Subprocess 3.4 or valid amount is sent to Subprocess 3.1, as appropriate for the particular transaction. If a limit is exceeded, an error message is displayed and the session is terminated.

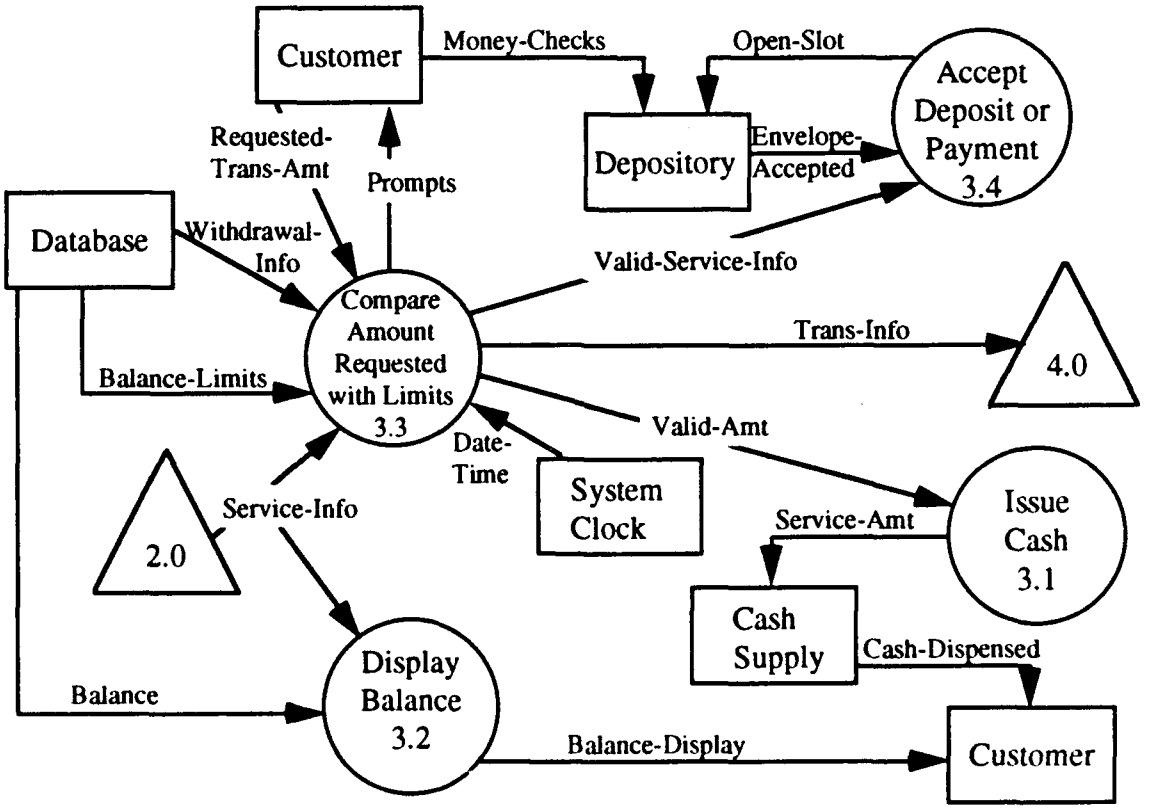


Figure 19. Process 3.0 - Process Transaction DFD (Loy, 1990:456)

Subprocess 3.4 - Accept Deposit or Payment receives an envelope accepted signal from the depository indicating that a deposit has been made. It also receives valid service information from Subprocess 3.3 containing information about the deposit. This

subprocess simply opens the deposit slot and prompts the customer for the envelope when this type of transaction is requested.

4.0 - Perform Accounting. This process, as with Processes 2.0 and 3.0, requires further breakdown. Subprocesses to calculate the balances, calculate statistics, and update limits were necessary (Figure 20).

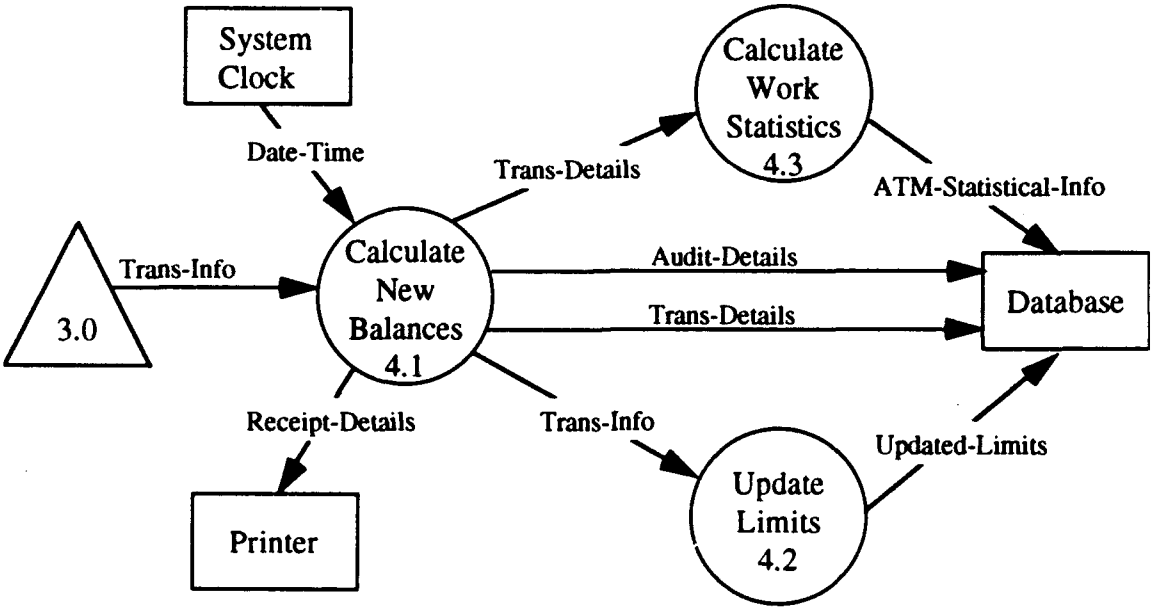


Figure 20. Process 4.0 - Perform Accounting DFD (Loy, 1990:456)

Subprocess 4.1 - Calculate New Balances receives the date and time from the system clock and the transaction information from Process 3.0. Depending on the type of transaction requested, this subprocess takes either no action or adds/subtracts the transaction amount to/from the appropriate balance. This subprocess outputs the receipt details to the printer, transaction information to Subprocess 4.2, transaction details and audit details to the database, and transaction details to Subprocess 4.3.

Subprocess 4.2 - Update Limits receives transaction information from Subprocess 4.1 and subtracts the requested transaction amount from the current limit to arrive at a new limit. The updated limit is then sent to the database.

Subprocess 4.3 - Calculate Work Statistics receives transaction details from Subprocess 4.1 and uses them to calculate the following statistics: the number of customer sessions, the number of transactions completed, and the number of each type of transaction completed. The following status information is also calculated: the amount of cash left, the number of customer receipts remaining, and the dollar amount expected in the depository. The subprocess outputs the ATM statistical information to the database.

Data Dictionary. A data dictionary (Table 4) is also used to track the composition of the pieces of data flowing from one process to another. In the interest of space, we will only see the data dictionary for the context diagram. The completed data dictionary can be found in the full AUTOTELLER specification in Appendix C.

Table 4
Partial AUTOTELLER Data Dictionary (Loy, 1990:452-454)

Data	Composition
Acct-details	= audit-details + ATM-statistical-info + trans-details
ATM-cust-ID-info	= PIN + ATM-card-number + cust-ID-number
Balances-Limits	= 1{balance}2 + service-limits
Cust-acct-list	= 1{acct-designation + acct-number}8
Cust-info	= cust-card-PIN + requested-trans + requested-trans-amt
Date-time	= date + time

Eject-receipt-signal	= *signal to the printer that the customer session is complete and the receipt should be printed*
Prompts	= *messages displayed on the terminal screen*
Receipt-details	= 1{completed-trans-info}4
Receipt-header-info	= time + date + location + cust-id-number
Service	= [cash-dispensed | deposit-accepted | loan-payment-accepted | balance-displayed]
Withdrawal-info	= date-last-withdrawn + acct-withdrawal-limit

Constraints/Nonbehavioral Requirements. The DFDs and data dictionary represent most of the users' (both bank customer and bank employee) behavioral requirements. However, some requirements do not fit into a DFD or data dictionary. We will represent some of these as constraints in the specification:

- The system must interact with the bank's existing central computer system.

In addition, several nonbehavioral requirements must also be addressed in the specification:

- The response time for menu changes should be no more than three seconds.
- The response time to read an ATM card should be no more than three seconds.
- Sixty seconds should be allowed for envelope insertion when a deposit is requested. For all other prompts, the customer should have thirty seconds to respond.

- The receipt should be issued no longer than two seconds after the session is terminated.

- Computational errors must never occur, and the system must never lock up due to customer errors.

- Adequate security safeguards will be built into the system.

- The system can be out of service no more than 0.001% of its yearly operating time. Regularly scheduled maintenance is not included.

The above context diagram, data flow diagrams, data dictionary, constraints and nonbehavioral requirements can then be used to construct a specification of the AUTOTELLER system which may be used to enter into preliminary design. The level of detail presented here is sufficient for the purposes of this research effort. The complete specification is attached as Appendix C.

SQFD Representation of the Sample Problem

We will now use the methods discussed in Chapter II to analyze the same sample problem, but this time we will use SQFD as an aid. All the requirements and constraints will remain the same as in the structured analysis decomposition in the interest of comparison.

User Analysis. One of the first steps in SQFD is to analyze exactly who the users are and how they relate to the system (Zultner, 1990:134). The users can come from many different areas and be interested in different aspects of the system. There are also stakeholders in the system. These stakeholders may not be operators of the system,

but may decide on important aspects such as policy, acceptance, funding etc. The operator of the system is not the only user in this context. In order to achieve our goal of getting the voice of the customer into the system, we must have an idea of exactly who the customers are and how they relate to the system.

Our first SQFD matrix, called the Z-0 matrix (Figure 21), will attempt to identify the different customers and their role in the system (Zultner, 1990:134). In the AUTOTELLER system we have two distinct groups of users; the *bank customers* and the *bank employees*. The first group (*bank customers*) can be divided into subcategories: *frequent users* of the ATM, *non-technical*ly oriented users, *occasional users*, and *new users*. The second group (*bank employees*) can be subdivided into: *maintainers*, *tellers*, *managers*, and the *owner*(s).

A user can be involved with the system from different, and sometimes unthought-of ways. Some users are external, some internal, some decide policy etc (Zultner, 1990:134). In the Z-0 matrix, the users are listed across the top of the matrix, and their interests are listed down the side of the matrix. Relationships are established in the resulting matrix using the symbols previously discussed in Chapter II.

In our example, the users' interests has been divided into two categories. The first describes operational interests such as: *easy to use*, *quick*, *capabilities*, and *available*. The other category describes non-operational interests such as: *policies*, *funding*, and *security*. These interests may be determined through interviews, surveys, focus groups, trouble reports etc (Zultner, 1990:134). The relationships between the users and the users' interests are determined and represented by the customary symbols. A double circle for a strong positive relationship (value of nine), a single circle for some relationship (value of three), and a triangle shows a possible relationship (value of one).

If no relationship exists, the matrix is left blank. Zultner suggests five new symbols with values of one, three, five, seven, and nine which he feels offer a more intuitive meaning (Zultner, 1990:140-141). For this research effort, the more widely accepted symbols and values will be used. Subjective values of the importance of the users' interests are also included. The final step is to solve the matrix by summing the product of the values of importance(s) and the relationship(s) and entering the value in the row beneath the matrix. The result is an absolute value of each users' importance or clout. A relative value is calculated for clarity.

		System Users / Stakeholders: User Characteristics	Importance	Users: Bank Customers: Frequent Users	Bank Customers: Non Technical Users	Bank Customers: Occasional Users	Bank Customers: New Users	Bank Employees: Maintainers	Bank Employees: Tellers	Bank Employees: Managers	Bank Employees: Owner
Stakeholders' Interests	Operational Interests	Easy to use	4	△	◉	○	◉		△		
		Quick	4	○			△		◉		
		Capabilities	4	◉	△	△	○		◉		
		Available	4	○			○		○		
	Non Operational Interests	Policies	1					△	△	○	◉
		Requirements	3	△	△	△	△	○	○	○	
		Funding	1								◉
		Approval Auth	1							○	◉
		Testing	4					○	○	△	
		Security	3	○	○	○	○		○	◉	
		Documentation	2					◉	○		
Users Absolute Importance				76	52	28	76	40	125	46	27
Relative Importance				16.1%	11.0%	5.9%	16.1%	8.5%	26.5%	9.7%	5.7%

Figure 21. AUTOTELLER Z-0 Matrix User Characteristics vs Users

For the AUTOTELLER system, the most important users are the *tellers* with *frequent users* and *new users* second. This may seem to be counter-intuitive, but serves to demonstrate the value of this simple analysis.

User Requirements. Now that we have an understanding of the relative importance of the users, we can begin to determine the users' requirements.

With the same techniques as those used above, we collect and review the users' requirements and arrange them in hierarchical fashion on the left side of the Z-1 matrix (Figure 22). We also include the importance values that we gathered from the user surveys. The Z-1 matrix relates the users to the users' requirements in order to place a priority on each user requirement based on the importance of the associated users (Zultner, 1990:135).

The requirements were broken down into two groups; those associated with the bank customers, and those associated with the bank employees. Further breakout followed the requirements identified in the beginning of this chapter. Once all the requirements were in place, relationships between the users' requirements and the users were determined. At the bottom of the matrix, the users' relative importance (multiplied by one hundred) was brought forward from the Z-0 matrix.

Once all the information is in place, the matrix can be solved. The users relative importance (at the bottom of the matrix) is multiplied by the values of the relationships, then added to arrive at the raw priority (on the right hand side of the matrix) of the user requirements. This raw priority is then multiplied by the importance value and the result is the adjusted priority of the user requirements with the user importance taken into account. Relative values are calculated for ease of understanding.

User Requirements / System Users / Stakeholders			Importance	Stakeholders: Bank Customers: Frequent Users	Bank Customers: Non Technical User	Bank Customers: Occasional Users	Bank Customers: New Users	Bank Employees: Maintainers	Bank Employees: Tellers	Bank Employees: Managers	Bank Employees: Owner	Raw Priority	Adjusted Priority	Relative Adjusted Priority
Bank Customers' Req	User Interface	Menu System	5	○	◉	○	○					213.3	1066	4.0%
		Friendly Prompts	5		◉	△	◉					249.8	1249	4.7%
		Responsive	5	◉	△	△	○		△			236.6	1183	4.5%
		Trans Receipt	4	○	○	○	○		○			226.8	907	3.4%
	Capabilities	Make Withdrawal	4	◉								144.9	579	2.2%
		Make Deposits	2	○	○	△						87.2	174	0.6%
		Check Balances	3	◉	◉	○	○					309.9	929	3.5%
		Transfer Funds	3	○	△	△	△					81.3	243	0.9%
		Pay Loans	1	○	△							59.3	59	0.2%
Bank Employees' Req	Accounting	Maintain Balances	5						◉			238.5	1192	4.5%
		Update Limits	2	△					△			42.6	85	0.3%
	Information Tracking	Collect Statistics	2					△	○	◉	○	192.4	384	1.4%
		Track Cash	4					○				25.5	102	0.3%
		Track Receipts	2					○				25.5	51	0.1%
		Verify ATM Card	5						○			79.5	397	1.5%
	Misc	*Available*	5	△				◉	◉			331.1	1655	6.3%
		Secure Accounts	5	○	○	○	○		○	◉	○	331.2	1656	6.3%
		No Comp Errors	5	△	△	△	△		○	◉	△	221.6	1108	4.2%
Users Relative Importance				16.1	11.0	5.9	16.1	8.5	26.5	9.7	5.7			

Figure 22. AUTOTELLER Z-1 Matrix User Requirements vs Users

Technical Requirements. Now that the user requirements and their priorities are understood, the technical requirements needed to meet those user requirements can be determined. The A-1 matrix (Figure 23) is used here. These requirements should be a measurable property of the system which can be manipulated to help meet the user's requirements (Sharkey, 1990:10).

The user requirements were brought forward from the Z-1 matrix and placed down the left hand side of the A-1 matrix. The adjusted priority from the Z-1 matrix (multiplied again by one hundred) is also brought forward. The technical requirements,

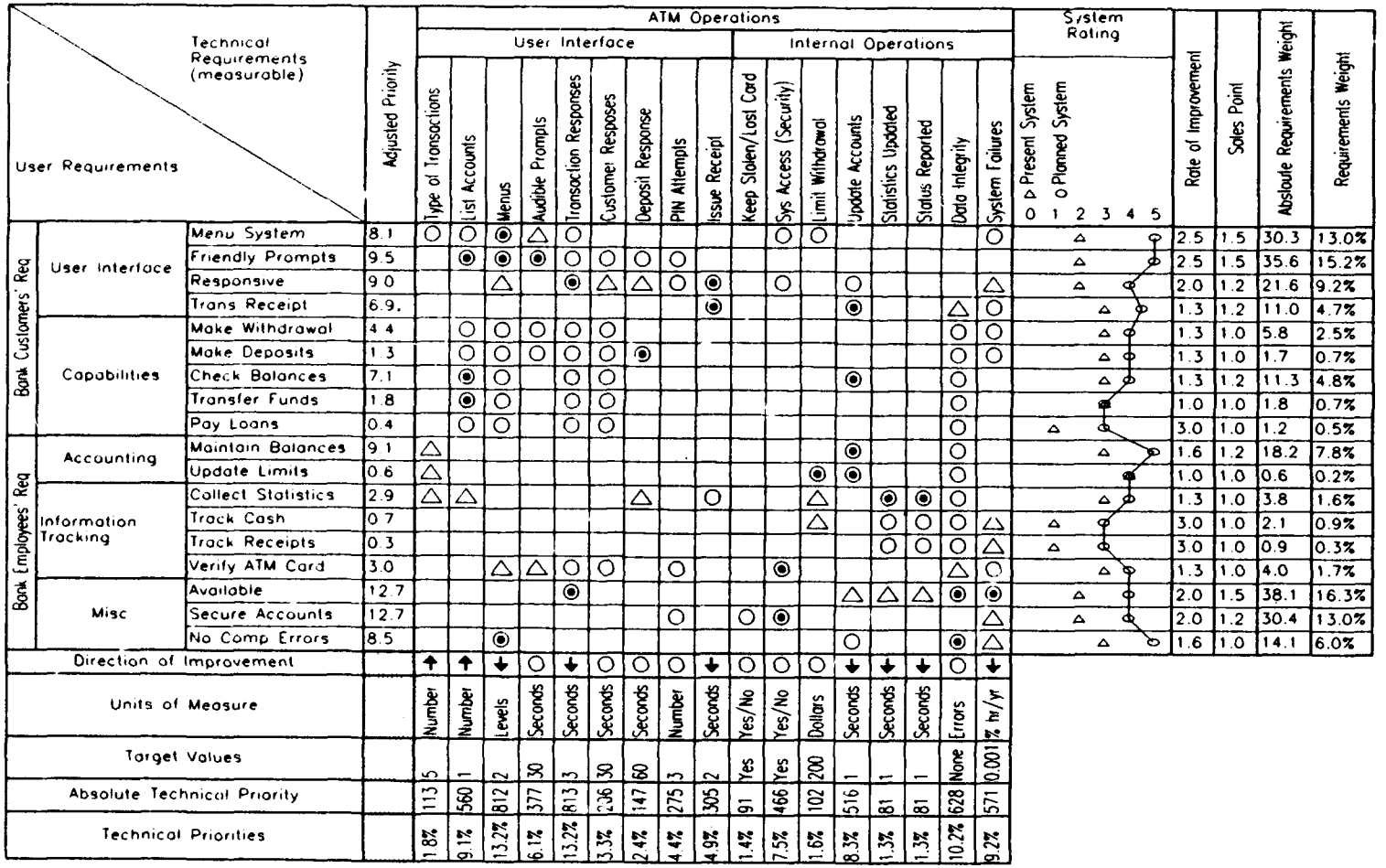


Figure 23. AUTOTELLER A-1 Matrix User Requirements vs Technical Requirements

placed across the top, were broken into two groups. Those related to the user interface include requirements for the number of transaction types, list of accounts, and system response (and wait) times. The technical requirements for the internal operations of the ATM include additional system response times, security requirements, and error allowance. For each technical requirement, a target value is identified. These values should be such that they may be verified at a later time. A direction of improvement may also be added. For example, the statistics should take no more than one second to be updated, the quicker the better. For PIN attempts however, a value of exactly three is assigned. This is symbolized by the open circle. Once the technical requirements have been established and reviewed, relationships are filled in using the same symbols as earlier.

The A-1 matrix includes an assessment of the present and planned systems. Testing of the present system may be used to assess its position, on a scale of one to five, on each of the user requirements (King, 1989:4-5). If we were in a competitive environment, the competitors' positions on the same users' requirements would also be evaluated. This information, combined with the adjusted priority, can help establish a plan for where the improvements to the new system should be emphasized. For example, since the present system only rates a two for *friendly prompts*, and *friendly prompts* has a relatively high adjusted priority, we will plan the new system to attain a five in this area. The rate of improvement is simply the ratio of the planned system rating to the present system rating (King, 1989:4-6).

The sales point is intended to apply added emphasis on a feature that could result in higher sales for a commercial product. As discused in Chapter II, the sales point can take on values of 1.5, 1.2, or 1.0, and are assigned sparingly (King, 1989:4-6). For the AUTOTELLER system, sales points were assigned to user requirements that were felt

would lure additional customers to the ATM. The sales points were placed against some of the highest priority user requirements.

The absolute requirements weight is determined by taking the product of the adjusted priority, rate of improvement, and sales point for each user requirement (King, 1989:4-6). The requirements weight is simply the percentage of the absolute requirements weight, making it easier to understand.

Now the matrix is solved using the same techniques as before, but this time multiplying the relationships by the absolute requirements weight. The resulting absolute technical priority (and the percentage values) identifies the priorities of the technical requirements based on the users priorities. This will again be brought forward to the next matrix.

Processes. The next step is to take the technical requirements identified in the A-1 matrix and model them with a software analysis model. If a structured analysis approach is desired, we would use process models and data models (Zultner, 1990:136). An object oriented approach may use objects and services as a basis for analysis. For this sample problem we will stay with the structured analysis approach as shown earlier in this chapter. For the sake of simplicity, and in order to better draw conclusions of the usefulness of SQFD, we will adopt the same data flow diagrams and entities previously developed for the AUTOTELLER system.

Across the top of the A-2 matrix (Figure 24) are the processes identified through our structured analysis. The processes are identified by their name and number. Down the left hand side are the technical requirements and their priorities as determined from the A-1 matrix. Relationships are determined and placed within the matrix as before. When the matrix is solved by multiplying the relationships by the technical importance,

we determine an importance value for each process. This now tells us the value to the user of each process in the system. A similar matrix can also be constructed for the data entities in the system, resulting in an importance rating for each data entity.

Technical Requirements (measurable) / Product Functions (processes)			Technical Priorities	ATM System Process												
				Verify Customer 1.0		Initiate Transaction 2.0			Process Transaction 3.0				Perform Accounting 4.0			
				Verify Card	Check Stolen/Lost	Begin Customer Session 2.1	Select Transaction Type 2.2	Select Accounts 2.3	Issue Cash 3.1	Display Balance 3.2	Compare to Limits 3.3	Accept Deposit/Payment 3.4	Calculate New Balances 4.1	Update Limits 4.2	Calculate Statistics 4.3
ATM Operations	Internal Operations	System Failures	9.2	○	△				○			○			
		Data Integrity	10.2	△					△	△	○	△	◉	◉	○
		Status Reported	1.3			△			◉			◉			
		Statistics Updated	1.3			○	○		△			△			◉
		Update Accounts	8.3						○	○	○	○	◉	○	
		Limit Withdrawal	1.6				△	○	◉		◉		△	◉	
		Sys Access (Security)	7.5	◉	◉	△								△	
		Keep Stolen/Lost Car	1.4	◉	◉										
	User Interface	Issue Receipt	4.9			○	○	△	○			○			○
		PIN Attempts	4.4	○	○	○									
		Deposit Response	2.4				○	○		△		◉	△		○
		Customer Responses	3.3	○		○	○	○		△		○			
		Transaction Responses	13.2	○	○	◉	○	○	○	○	○	○	◉	◉	
		Audible Prompts	6.1	○		○	○	○	◉	○		◉			
		Menus	13.2	○		◉	◉	◉	△	○		○			
		List Accounts	9.1			△	○	◉		◉					
		Types of Transactions	1.8			△	◉	○	△	△		△			
Absolute Importance				238	142	317	257	290	214	222	109	257	289	257	64
Relative Importance				8.9%	5.3%	11.9%	9.6%	10.9%	8.0%	8.3%	4.1%	9.6%	10.8%	9.6%	2.4%

Figure 24. AUTOTELLER A-2 Matrix Technical Requirements vs Product Functions

The results of the SQFD process is a structured analysis representation of the system which has been developed based on the users' priorities of the various attributes of the system. These priorities were determined initially from the Z-0 matrix and flowed down through to the A-2 matrix, ultimately assigning a priority to each process in the system. We can now see a value to the user of any process, *begin customer session* for example, and know what importance to place upon it. The matrices also provide a level of traceability from the users' original desire for *easy to use* (for example) through the users' requirements, technical requirements, and finally to the processes needed in implement this vague user need.

This is as far as this sample problem will be studied. The SQFD process may be carried out further as suggested in Chapter II; however, the representation in this chapter is all that is necessary for this research effort.

V. Findings and Analysis

Introduction

This chapter presents the results of the survey described in Chapter III. The survey was distributed to twenty-two knowledgeable personnel, of which fourteen were returned. The breakout of these personnel were as follows:

- four to Academia
- four to software acquisition managers
- ten to software engineers
- and four to user organizations

The breakout of the survey responses were as follows:

- four from Academia
- four from software acquisition managers
- four from software engineers
- and two from user organizations

The raw data was transposed and consolidated for the sake of readability and is presented for review in Appendix B.

Summary of Survey Results

All nine of the survey questions are listed below along with a summary of the respondents' replies. Since the sample size was so small, no statistical methods were employed to analyze the responses.

1. Do you think that the structured analysis representation of the sample problem (not taking into account the SQFD analysis) adequately captures all the users' wants, needs, and desires? Please explain.

The goal of this question was to determine if the respondents felt structured analysis captured those user needs that may not come out as clear cut requirements; i.e., things the user wanted but did not really know how to state them.

Eight of the respondents replied that the structured analysis did not capture all of the users' requirements. Some of the respondents analyzed this question in too much detail, citing specific requirements that were not represented. This was not the goal of the question. One respondent pointed out the DFDs can only model certain type of user requirements, while some were of the opinion that, when properly performed, structured analysis could adequately capture the users requirements.

2. Does the SQFD approach help to capture any of those needs, wants, desires, you felt were missing (if any) from the Structured Analysis approach? Please explain.

The goal of this question was as a follow-on to the previous question. Given that the structured analysis failed to capture those more nebulous requirements, did SQFD help to find them. For example, did SQFD help to capture desires such as *user friendliness, easy to use* etc.

Ten respondents answered this question positively. Only two respondents answered negatively, while two others were undecided. One respondent who answered yes to this question was not sure how useful the additional information would be for system developers. Another respondent felt it was a much more analytical method of

determining user requirements. One respondent thought that SQFD would help to identify needs that were not captured by structured analysis.

3. Key to the SQFD process is capturing the importance (or priority) of the various characteristics of the problem. Do you believe that this is missing from most Structured Analysis representations? Is this useful information to determine?

The goal of this question was to determine if any of the respondents felt that structured analysis does or does not provide a mechanism for prioritizing requirements. This was identified as a shortcoming of the structured analysis method.

The responses to this question were overwhelmingly positive. Nine respondents felt that structured analysis did not provide a means to determine priority, while the rest either did not address this specifically in their answer, or simply discussed the importance of priority in general. No respondents answered that structured analysis did provide this capability. Thirteen of the respondents stated that the priorities are beneficial to have, with adjectives like "essential," "very useful," and "critical." One respondent did not address this concern.

4. In your opinion, would the use of SQFD provide a better means of requirements traceability then methods you currently employ? Please explain.

The goal of this question was to determine if using SQFD would provide a means of tracing user requirements. This was also identified as a shortcoming of structured analysis.

Nine of the respondents answered this question positively. Three stated that traceability would not be enhanced and two did not answer the question directly. Some of the positive answers suggest that it is an effective way of tracing the priorities of the

requirements. One response from a user suggested that without proper training SQFD might confuse users more than it would help them. Another respondent stated that SQFD could provide a means of tracing requirements from the SORD (System Operational Requirements Document) level. Lastly, one respondent cautioned that SQFD should be implemented early in the program. This was felt necessary to avoid re-identifying requirements that have already been deleted from a previous requirements analysis, preventing conflicts between the user and buyer.

5. The SQFD Z-0 matrix attempts to analyze the different users and their relative importance or clout. From your experience, do you believe that this would be useful information? Please explain.

The goal of this question was to determine if the Z-0 matrix, peculiar to the Zultner methodology, was useful or not. If the Z-0 matrix was not considered useful, it could be deleted from future USAF models.

Nine respondents answered this question positively, four did not give direct answers, and only one responded negatively. There was some concern as to whether or not the user importance reflected the real importance and whether less important users would be offended by this status. One respondent made the comment that this information could help in building a system incrementally; determining which capabilities should be implemented first. One respondent felt that in the military environment some requirements are actually absolute and therefore must be delivered. In other words, some requirements are not negotiable and therefore are not subject to a weighting. Lastly, one respondent saw this information as being helpful in identifying if difficult requirements should be pursued or not.

6. Do you believe that the additional information resulting from using SQFD could result in a better Structured Analysis of the problem? In other words, would SQFD provide a better front end to Structured Analysis then present analysis methods? Please explain.

The goal of this question was to see if SQFD would fit into the structured analysis methods and provide useful information that may not otherwise be captured.

Nine respondents answered positively, four did not directly answer the question or felt there was too little information to base an answer on, and only one responded negatively. One respondent stated that SQFD would help "bridge the gap" between the users' requirements and structured analysis. Another felt that SQFD would help to ensure that the important requirements would be captured, while structured analysis would help to ensure completeness.

7. Based on your experience, are there any reasons why SQFD would not be "workable" in the USAF environment? Are there obstructions that would not make SQFD possible to use?

The goal of this question was to try to identify roadblocks to the adoption of SQFD. Later SQFD models could attempt to correct any deficiencies identified.

No serious reasons were given would block the adoption of SQFD in the USAF environment. Proper training to use and understand SQFD was considered important. A concern was raised that SQFD could lead to the committee approach to design that would please everyone a little, but no one a lot. The most serious concern was how to use SQFD for large and complex systems. In other words, it was not clear how SQFD would scale up beyond the simple sample problem presented. Two respondents identified the

need to convince contractors to become involved in SQFD or it would not be of value to the USAF. Finally, it was felt important to implement SQFD early in the program cycle.

8. If you were trained to use SQFD, would you seek to apply it to a real-world program?

The goal of this question was to gauge the respondents acceptance of the SQFD methodology.

Ten respondents answered this question positively, one needed more information to make a determination, and two responded negatively, one of which would not use it in its present form. One respondent identified the need to have an automated tool so that the process would be quicker and would not be ignored due to the difficulty of constructing its complicated matrices. Another respondent saw SQFD as more of a configuration management tool rather than a front end to structured analysis, but did not explain why he felt this way.

9. Any additional comments you may have would be greatly appreciated, including any suggested modifications to the SQFD process.

The goal of this question was to identify any concerns that the respondents may have had, but found no other opportunity to state them. Suggested modifications to the SQFD process were also looked for so that future SQFD models could incorporate them.

Responses to this question were, of course, wide and varied. One of the issues raised was the use of SQFD on large and complex systems. It is not obvious how SQFD would scale up to these large problems. One respondent suggested a responsibility matrix be added to SQFD to identify who is responsible for each requirement. It was felt that SQFD made people think requirements out more thoroughly and concentrate effort

on the most critical aspects of the requirements. On the other hand, it is still a subjective measure and gives basically subjective (as opposed to a formal mathematical method) results. One respondent wanted SQFD to allow for absolute requirement; i.e., one with an infinite weighting. Lastly, another felt the survey itself was too subjective and was difficult to answer.

Analysis with Respect to Original Questions

The literature review, tailored QFD process, and survey results will be analyzed with respect to the investigative question.

Investigative Question 1. What are the QFD process, its goals, products and techniques?

The purpose of this question was to determine the present QFD process in order to better understand how the process may be tailored to the software development model.

A review of present literature on the subject of QFD was accomplished and can be found in Chapter II. The QFD process seeks to integrate the voice of the user into the design of the product. QFD accomplishes this goal through a series of matrices designed to identify the customers' real needs and desires and prioritize them to efficiently produce the product.

Investigative Question 2. What are the current techniques available to develop software systems requirements for the USAF?

The purpose of this question was to determine the present methods and techniques of requirements analysis and their shortcomings. It was these shortcomings that the tailored approach to QFD was intended to help overcome.

The method of requirements analysis identified was the structured analysis technique. This included data flow diagrams, data dictionaries, and entity-relationship diagrams. Three shortcomings were identified in the structured analysis method. First the structured analysis still does not capture the users' requirements adequately. Second, structured analysis does not provide any mechanism for capturing the priority of the various user requirements. Lastly, the structured analysis does not provide a mechanism to trace the users requirements through to the structured analysis representation.

Investigative Question 3. Can the QFD process be tailored to meet a specific domains' requirements?

The purpose of this question was to determine if the QFD process can be tailored to meet specific requirements, or if the QFD process is domain specific; i.e., to the automotive industry.

Chapter II identifies a software peculiar method of QFD called SQFD for this research effort. One document SQFD model was discussed, and this model was further modified by another undocumented SQFD model. In addition, several instances of the use of SQFD in industry were reviewed.

Investigative Question 4. What specific tailoring of the QFD process should be made to use QFD to aid in developing software systems requirements in the USAF environment?

The purpose of this question was to determine the specific methods to use in applying QFD in the software development domain. These specific methods would then be used to develop a sample problem which could be compared to a structured analysis representation of the same problem.

The specific tailoring of the QFD process was identified in Chapter II. These methods were very similar in concept to the standard QFD model, but added some matrices and modified other matrices to meet the needs of software vice hardware. These methods were further used to construct a sample problem in Chapter IV.

Investigative Question 5. Is the QFD process acceptable to USAF software managers/engineers?

The purpose of this question was to determine if the QFD process can be used in the USAF environment. If the QFD process does show promise of providing benefits to the area of software requirements analysis, can the process fit into the USAF way of doing business.

The consensus of survey respondents indicates there does not seem to be any great impediment to using SQFD in the USAF environment. The greatest obstacle was the use of SQFD on large systems. The respondents were not sure how well SQFD would scale up to a larger and more complex system than the sample problem provided. In the automotive application of QFD, this is handled through hierarchical techniques. Similar techniques could be used in SQFD as well. Training was deemed to be important so that both those implementing SQFD, and the users, would feel comfortable with the SQFD process and accept its results.

Investigative Question 6. Does the application of the tailored QFD process result in requirements analysis that correct shortcoming of current techniques?

The purpose of this question was to determine if the application of QFD can overcome some of the shortcomings of the current techniques in requirements analysis as identified above. These shortcomings were: structured analysis still does not capture the

users' requirements adequately, structured analysis does not provide any mechanism for capturing the priority of the various user requirements, and structured analysis does not provide a mechanism to trace the users requirements through to the structured analysis representation.

The consensus of survey respondents was that SQFD does help in identifying user requirements better than the structured analysis method. Also, the respondents felt that structured analysis did not provide a means of prioritizing the users' requirements, and furthermore, this prioritization is useful information. Finally, the respondents felt that SQFD does provide some capability to trace the users' requirements, although this capability was not perceived to be robust.

VI. Conclusions and Recommendations

Significance of Results

The results of this research indicates that SQFD does show promise in overcoming some of the perceived shortcomings of the structured analysis method. In addition, the USAF environment does not appear to block the adoption of SQFD as a requirements analysis method.

In Chapter I we saw how important software is becoming in both private industry, and more specifically, the DOD. We also saw how errors inserted in the early phases of software systems development can lead to costly fixes later in the system life cycle. Chapter II provided evidence that many users of both QFD and SQFD have achieved savings in both time and money while simultaneously better satisfying their customers. If SQFD could reduce the amount of errors in requirements, it may save time and money for the USAF as well. If documented cases in industry are any indication of the gains to be realized in the DOD, these saving could well be significant, and result in systems that better meet our users' needs and expectations.

Practical Implications of the Results

While SQFD may have the potential to save time and money and better satisfy users, it is not a trivial matter to implement. The case studies in Chapter II are all from industry. A common thread throughout these cases is that the users of SQFD received proper training and there was a sincere motivation on the part of the participants and management to improve their products. This product improvement comes from a desire to increase the bottom line (or profits) of their companies. This same environment does

not exist in the USAF. Most, if not all, USAF personnel would sincerely like to improve the systems they work on, but the motivation that comes from the chances of being put out of business just do not exist. In addition, it is feared that USAF organizations may jump into SQFD without the proper training and guidance, only to see questionable results and determine that SQFD is not a viable tool.

A limitation of these results stems from the methodology. The purpose of this research effort was to determine if SQFD is feasible for the USAF and is worth investigating further. The survey methodology was subjective in nature in an attempt to gather input on SQFD rather than a head count of respondents answering yes/no type questions. The survey size was also small; only fourteen personnel responded to the survey. While these respondents were considered to be knowledgeable individuals, they cannot be considered representative of the entire community of USAF software personnel. Therefore, the results presented here may not be considered indicative of the attitudes of all USAF software personnel. The results do indicate however, that SQFD does have the potential to be beneficial and does warrant further investigation.

The responses to the survey were generally positive. The respondents felt that SQFD could help in identifying users' requirements, aid in requirements traceability, and provide valuable information on the priority of the users' requirements. While this study only attempted to determine the acceptability of SQFD in the USAF environment, more research is required to actually utilize SQFD.

Recommendations for Follow-on Research

This study was intended to determine if SQFD is feasible for use in USAF software acquisition and warrants further investigation. Based on this research, the

method clearly shows promise. However, more work remains before SQFD can be adopted for widespread use in the Air Force. The following course of research is recommend to fully validate SQFD.

- Develop a small application using the traditional structured analysis technique and the SQFD aided structured analysis technique. The application could then be examined by the users to determine which application better suited their needs. Metrics could also be gathered to help determine if time and effort are truly saved.

- Integrate additional software requirements methodologies into the SQFD model. Object Oriented Requirements Analysis and design would be one such methodology.

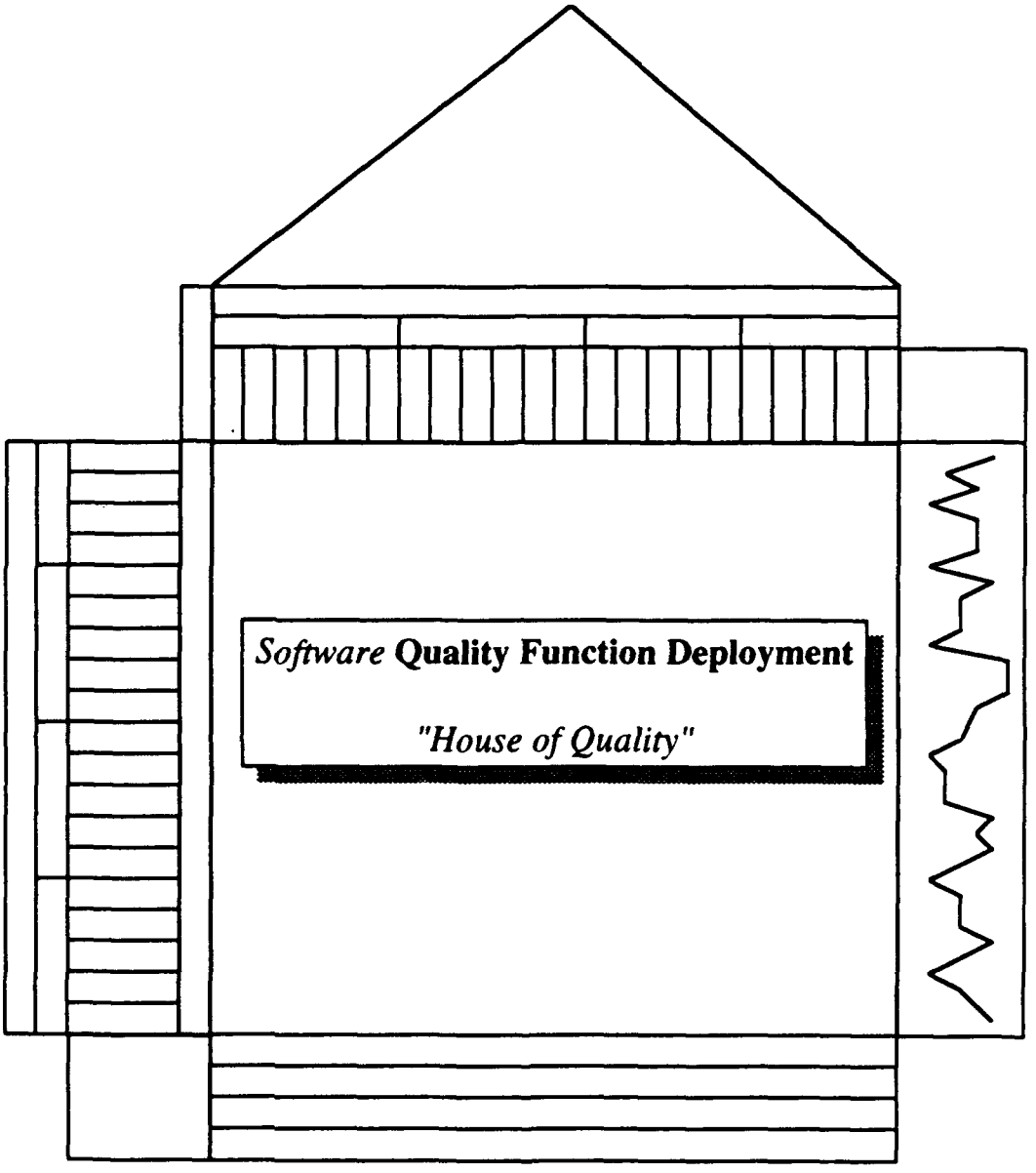
- Incorporate additional SQFD matrices into the model that would carry the methodology beyond the requirements analysis phase and through to the final product. This would culminate in a life cycle model which integrates SQFD into the software development process.

Conclusion

In this research effort, Quality Function Deployment and its application to software, SQFD, was investigated for potential use in USAF software acquisition. A survey was conducted to gather the opinions of software knowledgeable personnel. The survey participants compared an SQFD based requirements analysis of a sample problem to one based on the traditional structured analysis. Based on the positive response to this survey, and the experiences of industry, SQFD clearly shows promise. In a time when requirements analysis are causing schedule difficulties and cost overruns, SQFD's

potential cannot be overlooked. The recommended follow-on research should be carried out so that SQFD's full potential can be realized.

Appendix A: SQFD Survey Package



Captain Craig Lamb, USAF
Air Force Institute of Technology/LSG
Wright-Patterson AFB Ohio 45433

Introduction

Thank you for taking part in this research project. I hope that you will find it both interesting and educational. This package of information is arranged in five parts.

- ❑ A background on Quality Function Deployment (QFD).
- ❑ A sample problem scenario.
- ❑ A structured analysis interpretation of the sample problem.
- ❑ A QFD aided interpretation of the sample problem.
- ❑ And a questionnaire.

The background information is provided to give you an orientation of the purpose of QFD. The sample problem scenario defines the problem in terms of the various requirements to be met. From those requirements we will step through a simple structured analysis of the problem. Then, as if starting from the requirements again, we will step through the QFD aided analysis of the problem (which also incorporates the structured analysis representation). The final step is the questionnaire.

All the information needed to respond to the questionnaire is included in this information package. It may be helpful to review the questionnaire before and during your review of this package so that the level of detail the questions are asking for is understood. I think you will find that the questions are relatively high level and do not require an in depth study of the problems.

Please take whatever time you feel is necessary to review the package and respond to the questions. If you need more information, or wish to ask me questions, I can be reached through any of the following means:

Home phone - (513) 433-8328 AFIT message center (513) 255-8989

E-Mail via the internet - clamb@hercules.afit.af.mil

I realize that the information presented here is new and may require face-to-face discussion. I will be happy to meet with you to discuss the package anytime of the day on Mondays, Wednesdays, and Fridays. Please contact me and we will set up a time. Off-base personnel can call at the numbers above and discuss the package on the phone if you wish.

When you have completed the questionnaire, please send it to me at the following address:

AFIT/LSG
ATTN: Capt Craig Lamb
Bldg 641
WPAFB Dayton OH 45433

Please try to return the questionnaires NLT 26 Jul 91. I will gladly pick up completed questionnaires from on-base personnel. Again, I greatly appreciate your participation in this project - Capt Craig Lamb.

Background on Quality Function Deployment

Quality Function Deployment (QFD) is only one of many activities that fit into the larger picture of Japanese total quality management philosophy. QFD is a planning process that began in the Mitsubishi Kobe shipyards in 1972 (Hauser, 1988:63). QFD is not solely a quality tool as its name would imply. The QFD process is intended to take a product (hardware) from initial concept though design and production ensuring that the end product meets the needs of the customer. The customers' needs play a key role in QFD. It is these needs, properly analyzed that decide which way the product will evolve. Indeed, the final production plans should be traceable back to the original customer demand. For example, a worker on a production line installs a light-weight part made from light-weight materials in order to meet the customers' demand for a product that "is easy to lift" from the initial QFD planning matrix.

Sample Problem Scenario

The sample problem will be based on the TELLERFAST software system which, together with the automatic teller machine (ATM) forms the AUTOTELLER Automatic Teller System. This new system is intended to replace the existing teller machines. The TELLERFAST software performs all of the functions necessary for the AUTOTELLER system to operate. These functions include:

- Accepting and validating the ATM cards.
- Supplying and responding to user menus.
- Issuing cash.
- Accepting deposits and loan payments.
- Transferring funds among accounts.

There are two categories of users of this system. Bank customers are those users that interact with the system to conduct a transaction. Bank employees are those users that interact with the system to perform maintenance, query the system, and recharge the teller machines with supplies including money. The next two sections describe the two users' requirements in narrative form.

Customer Oriented Requirements. A survey was conducted of 250 regular users of automatic teller machines (ATMs). The goal of this survey was to better understand the customers' use and expectations of ATMs in order to produce a better system. The users were asked to rate the importance of each feature on a scale of one to five with five being the highest. The survey resulted in the following findings:

- A majority of the customers liked a menu system to prompt them through an ATM transaction. This feature was given a five.

- Most customers appreciated the use of audible signals to remind them to remove their money, receipt, and ATM card. This feature was also rated a five.

- The minimum types of transactions the customers expected were as follows along with their importance rating.

    - Balance inquiry - three
    - Deposit checks - two
    - Transfer funds between accounts - three
    - Loan payments - one
    - Withdrawal of cash - four

- An informative receipt should be issued at the end of each transaction. The receipt should have, as a minimum, the date, the account number, the amount of the transaction, and the finishing account balance. This feature was assigned a value of four.

In addition to the survey, 15 volunteer customers were used to conduct a assessment of the response times desired for the ATM operation. The customers used a specially modified ATM under the supervision of the ATM contractor. The results of this study were:

- The response time for menu changes should be no more than 3 seconds.
- The response time to read an ATM card should be no more than 3 seconds.
- Sixty seconds should be allowed for envelope insertion when a deposit is requested. For all other prompts, the customer should have thirty seconds to respond.
- The receipt should be issued no longer than 2 seconds after the session is terminated.
- Response times in general were felt to be important and were rated a five.

Bank Employee Oriented Requirements. Requirements from the bank employees were gathered through face-to-face discussions with employees of the bank including tellers, bank managers, and ATM maintenance personnel. The following list comprises all the bank employees' requirements. As before, each feature's importance was rated on a one to five scale.

- The system is required to interact with the bank's existing central computer system. This is not negotiable (and is really a constraint), therefore we give it a five.
- The system must verify the customer's ATM card and Personal Identification Number (PIN) are correct. Three chances will be given to correct the PIN. A list of stolen and lost cards will be maintained. Each ATM card number will be checked against this list for a match. Any lost or stolen cards will be kept by the machine and the customer informed. This feature was given a value of five.
- If an ATM machine cannot dispense cash due to insufficient funds in the machine, the session should be terminated with an error message and the machine will be put on stand-by. This was also given a four.
- The system will maintain information on the last withdrawal and will not allow more than $200 to be distributed in any given day. An error message will be issued in these instances. This feature is only rated a two.

- The system will perform the following accounting functions as a minimum:
  - Maintain account balances of the customer's accounts. Very important and rated a five.
  - Maintain the customer's deposit limit, credit card limit, daily withdrawal limit, and account withdrawal limit. These limits will be changed to reflect the customer's withdrawals. Not as important and rated a two.
- The following statistics and status information will be maintained. The importance ratings follow.
  - Number of customer sessions - two.
  - Number of transactions completed - two.
  - Number of each transaction type completed - two.
  - Current amount of cash left - four.
  - Number of customer receipts remaining - two.
  - Amount of cash dispensed - four.
  - Dollar amount expected in the depository vault - four.
- Computational errors must never occur, and the system must never lock up due to customer errors. This was considered very important and was rated a five.
- Adequate security safeguards will be built into the system. This is also considered very important and rated a five.
- The system can be out of service no more than 0.001% of its yearly operating time. Regularly scheduled maintenance is not included. This is also rated a five.

Structured Analysis Representation of the Sample Problem

Both of the above sets of requirements (from the bank customers and from the bank employees) were used to arrive at the structured analysis representation that follows.

AUTOTELLER Context Diagram. The first step in decomposing the system is to construct a context diagram. The context diagram for the AUTOTELLER system shows four external entities or terminators and the central transform which represents the ATM system.

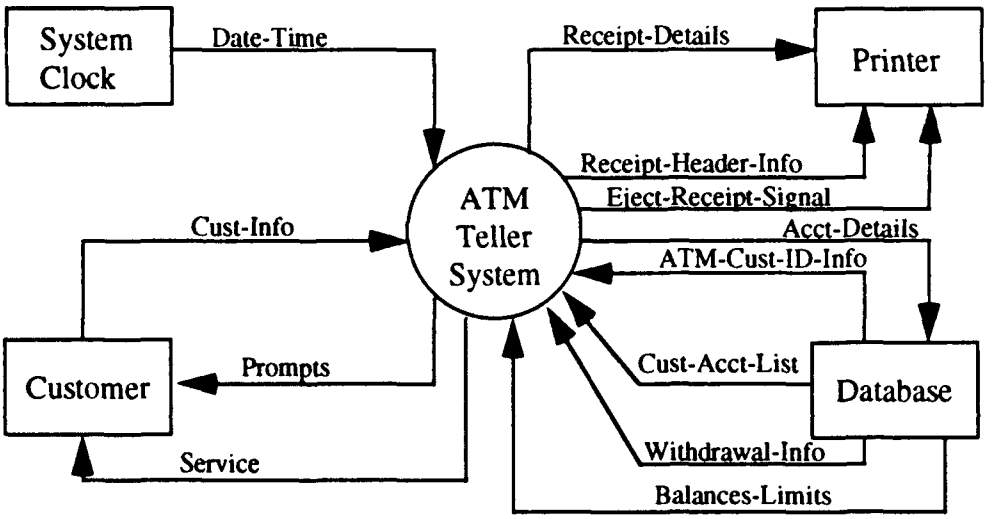


Figure 1. AUTOTELLER Context Diagram (Loy, 1990:454)

Each external entity was felt necessary to properly separate the system into logical components. The data flows between these components make up the total of all information needed at this level to satisfy the various needs of the customers.

- The *System Clock* supplies the time and date information.

- The *Customer* provides the system with information such as his/her PIN, a requested transaction, and the transaction amount, and is supplied with prompts from the system requesting action of the customer.

- The *Printer* prints and dispenses a receipt for a transaction.

- The *Database* contains all the information that the systems uses to accomplish a transaction.

❑ The *ATM Teller System* accomplishes all processing necessary for the AUTOTELLER system.

Data Flow Diagrams. The central transform of the context diagram does not contain sufficient information to design the AUTOTELLER system. The central transform "bubble" is therefore expanded to include more details of its internal processing. This top level data flow diagram is shown below. The central transform has been expanded into four numbered processes: Verify Customer ID, Initiate Transaction, Process Transaction, and Perform Accounting.

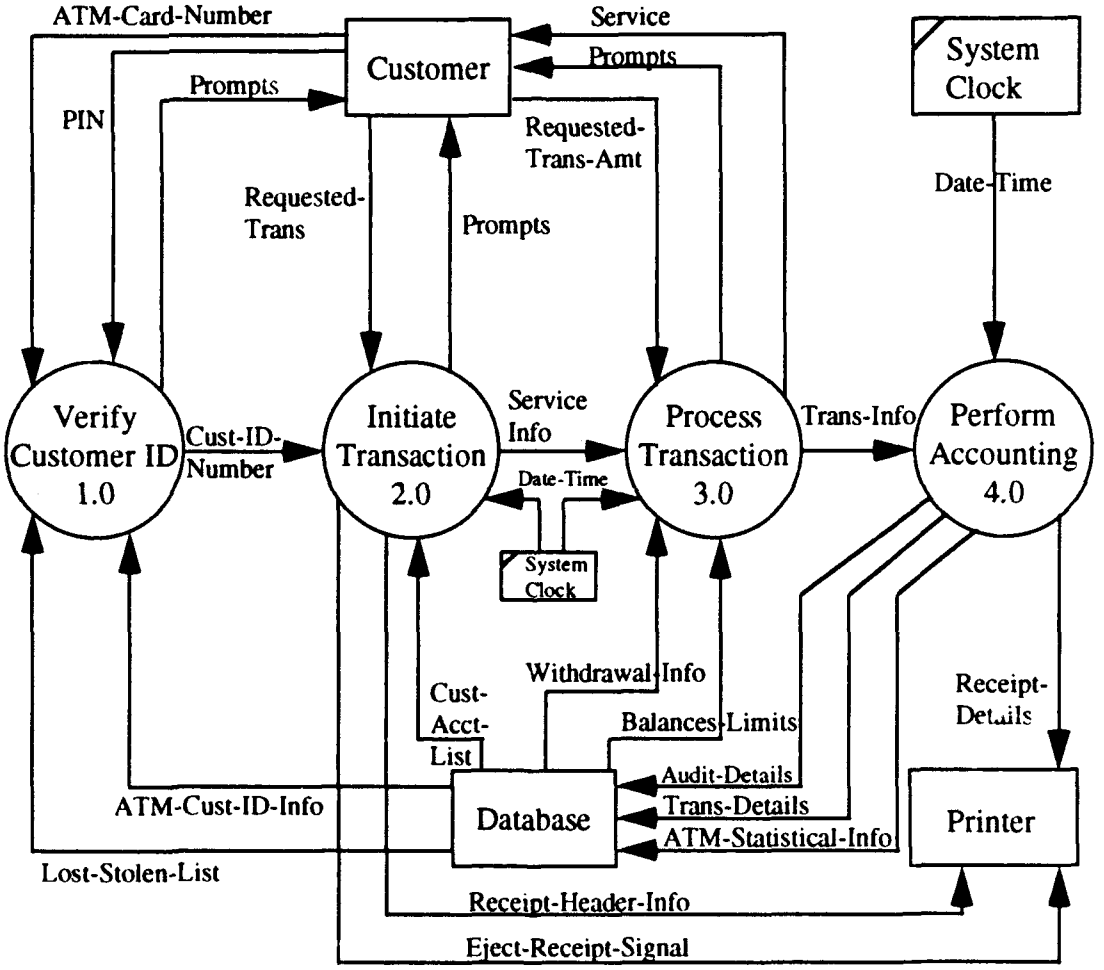


Figure 2. Top Level DFD (Loy, 1990:455)

This top level DFD helps to better define the system, but still more detail is needed to adequately define the system for a preliminary design. The top level DFD will therefore be further "leveled" to a second level set of DFDs. The first process is simple enough that no DFD is needed to describe its processing. However, the other three processes will require individual DFDs.

1.0 - Verify Customer ID. This process receives the card number and PIN from the customer as well as lost/stolen card information and customer information from the database. It is responsible for comparing the card number to the lost/stolen list and to retain the card if a match is found. This process must also prompt the customer for his/her PIN and verify the PIN against the database for correctness. The process passes the customer ID number to the Initiate Transaction process.

2.0 - Initiate Transaction. As shown in figure 4, this process had to be further broken down into three subprocesses.

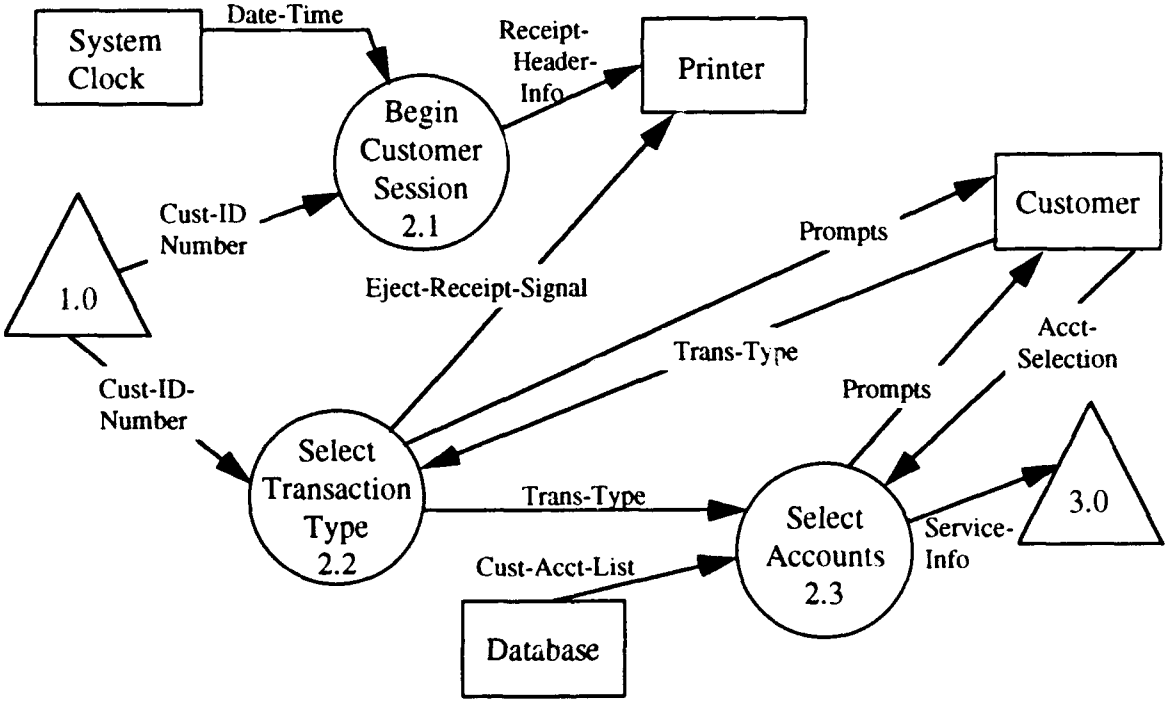


Figure 3. Process 2.0 - Initiate Transaction DFD (Loy, 1990:455)

Subprocess 2.1 - Begin Customer Session receives the customer ID number from process 1.0 (as represented by the numbered triangle) and the date/time from the system clock. This process outputs data to be printed at the top of the customer's receipt to the printer.

Subprocess 2.2 - Select Transaction Type receives the customer ID from process 1.0 and the transaction type from the customer. This subprocess prompts the customer for the transaction type and allows four transaction to be processed during one session. If the transaction chosen is not terminate, then the transaction type is sent to subprocess 2.3 When four processes have been performed, or the customer terminates the session, the eject receipt signal is sent to the printer to eject the customer's receipt.

Subprocess 2.3 - Select Accounts receives the account selection from the customer, the transaction type from subprocess 2.2, and the customer account list from the database. It uses this information and sends prompts to the customer to determine which account the customer wishes to affect. Depending on the transaction type there may be one or two accounts affected. The process outputs service information to process 3.0.

3.0 - Process Transaction. This process is the primary process of the top level DFD. It also required further definition, and was broken down into four subprocesses.

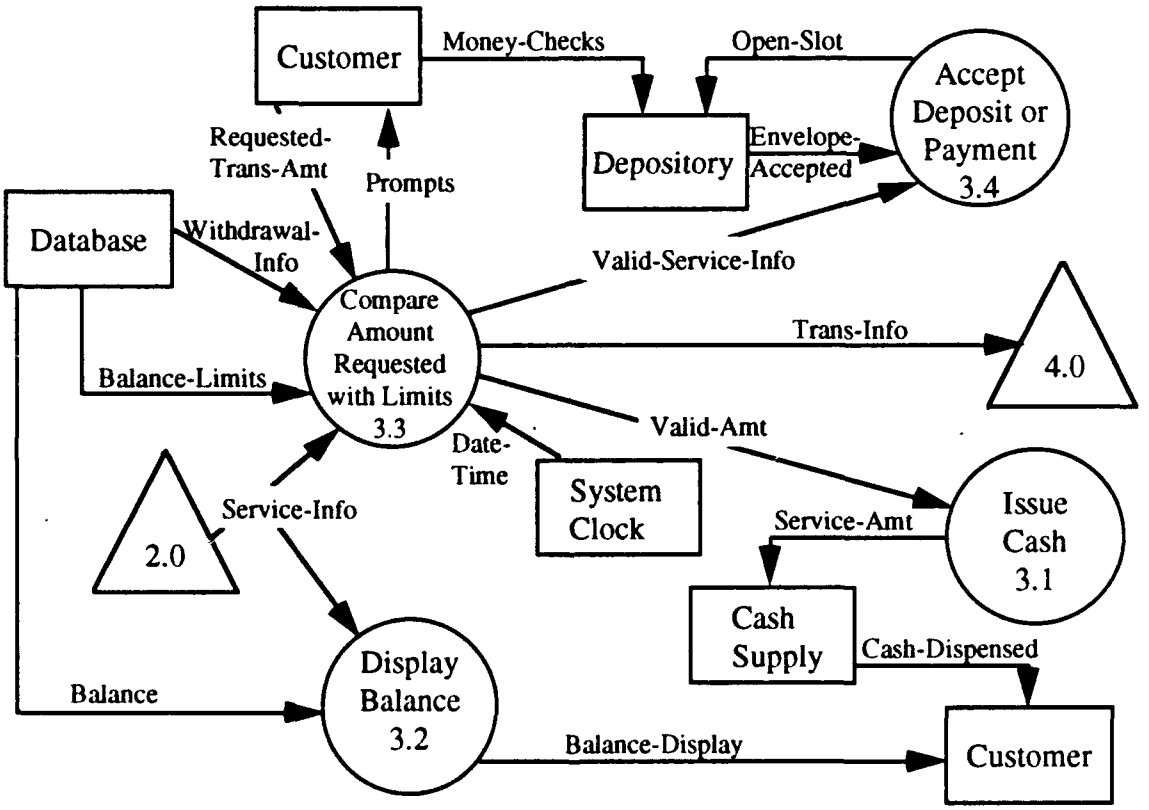


Figure 4. Process 3.0 - Process Transaction DFD (Loy, 1990:456)

Subprocess 3.1 - Issue Cash receives a valid amount from subprocess 3.3 and checks the supply of funds in the machine. If sufficient cash is available the service amount is sent to the cash supply for dispensing to the customer. If sufficient cash is not available an error message is displayed and the machine is put on standby.

Subprocess 3.2 - Display Balance receives service information from process 2.0 and the current balance from the database. The account designation and its balance are displayed to the customer.

Subprocess 3.3 - Compare Amount Requested with Limits receives withdrawal information, balance and account limits from the database, service information from process 2.0, transaction amounts from the customer, and the date and time from the system clock. This subprocess checks to see if the customer has withdrawn funds already this day, and if the funds withdrawn earlier this day exceeds $200. If not then the customer is allowed to withdraw funds that do not exceed the customer's limits. The customer is prompted for the transaction amount. After processing, the valid service information is sent to subprocess 3.4, or valid amount is sent to subprocess 3.1 as appropriate for the particular transaction. If a limit is exceeded, an error message is displayed and the session is terminated.

Subprocess 3.4 - Accept Deposit or Payment receives an envelope accepted signal from the depository indicating that a deposit has been made. It also receives valid service information from subprocess 3.3 containing information about the deposit. This subprocess simply opens the deposit slot and prompts the customer for the envelope when this type a transaction is requested.

4.0 - Perform Accounting. This process, as with processes 2.0 and 3.0, requires further breakdown. Subprocesses to calculate the balances, calculate statistics, and update limits were felt necessary.

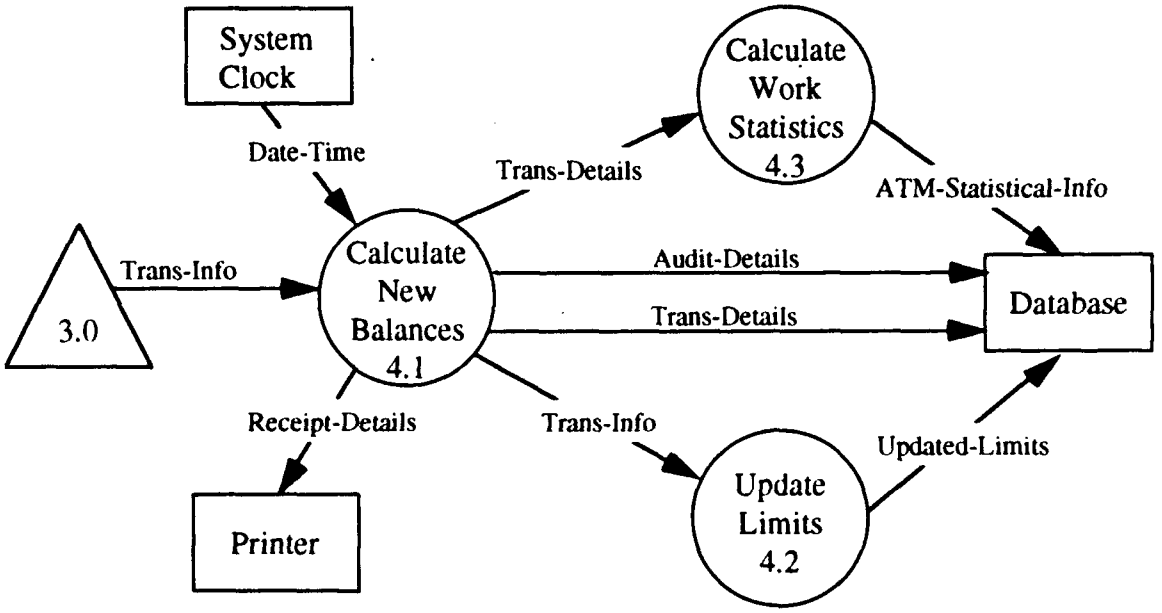


Figure 5. Process 4.0 - Perform Accounting DFD (Loy, 1990:456)

Subprocess 4.1 - Calculate New Balances receives the date and time from the system clock and the transaction information from process 3.0. Depending on the type of transaction requested, this subprocess takes either no action, or adds/subtracts the transaction amount to/from the appropriate balance. This subprocess outputs the receipt

details to the printer, transaction information to subprocess 4.2, transaction details and audit details to the database and transaction details to subprocess 4.3.

Subprocess 4.2 - Update Limits receives transaction information from subprocess 4.1 and subtracts the requested transaction amount from the current limit to arrive at a new limit. The updated limit is then sent to the database.

Subprocess 4.3 - Calculate Work Statistics receives transaction details from subprocess 4.1 and uses them to calculate the following statistics: the number of customer sessions, the number of transactions completed, and the number of each type of transaction completed. The following status information is also calculated: the amount of cash left, the number of customer receipts remaining, and the dollar amount expected in the depository. The subprocess outputs the ATM statistical information to the database.

Data Dictionary. A data dictionary or entity-relation diagrams can be used to analyze the pieces of data flowing from one process to another. Because the data flows are not germane to this research project, we will not study them further.

Constraints/Nonbehavioral Requirements. The DFDs and data dictionary represent most of the users (both bank customer and bank employee) behavioral requirements; however, some requirements do not fit into a DFD or data dictionary. We will represent some of these as constraints in the specification:

- The system must interact with the bank's existing central computer system.

In addition, several nonbehavioral requirements must also be addressed in the specification:

- The response time for menu changes should be no more than 3 seconds.
- The response time to read an ATM card should be no more than 3 seconds.
- Sixty seconds should be allowed for envelope insertion when a deposit is requested. For all other prompts, the customer should have thirty seconds to respond.
- The receipt should be issued no longer than 2 seconds after the session is terminated.
- Computational errors must never occur, and the system must never lock up due to customer errors.
- Adequate security safeguards will be built into the system.

- The system can be out of service no more than 0.001% of its yearly operating time. Regularly scheduled maintenance is not included.

The above context diagram, data flow diagrams, data dictionary, constraints and nonbehavioral requirements can then be used to construct a specification of the AUTOTELLER system which may be used to enter into preliminary design. The level of detail presented here is sufficient for the purposes of this research project. This completes the structured analysis representation.

SQFD Representation of the Sample Problem

We will now use SQFD methods to once again analyze the same sample problem. All the requirements and constraints will remain the same as in the structured analysis decomposition in the interest of comparison.

User Analysis. One of the first steps in SQFD is to analyze exactly who the users are and how they relate to the system (Zultner 1990:134). The users can come from many different areas and be interested in different aspects of the system. There are also stakeholders in the system. These stakeholders may not be operators of the system but may decide on important aspects such as policy, acceptance, funding etc. The operator of the system is not the only user in this context. In order to achieve our goal of getting the voice of the customer into the system, we must have an idea of exactly who the customer(s) is and how they relate to the system.

Our first SQFD matrix, called the Z-0 matrix, will attempt to identify the different customers and their role in the system (Zultner, 1990:134). In the AUTOTELLER system we have two distinct group of users, the bank customers and the bank employees. The first group (bank customers) can be divided into sub categories: frequent users of the ATM, non-technical users, occasional users, and new or potential users. The second group (bank employees) can be subdivided into: ATM maintainers, tellers, managers, and the owner(s).

In the Z-0 matrix, the users are listed across the top of the matrix, and their interests are listed down the side of the matrix. Relationships are established in the resulting matrix using the symbols shown in the legend in the figure below. These same symbols we be used throughout this sample problem.

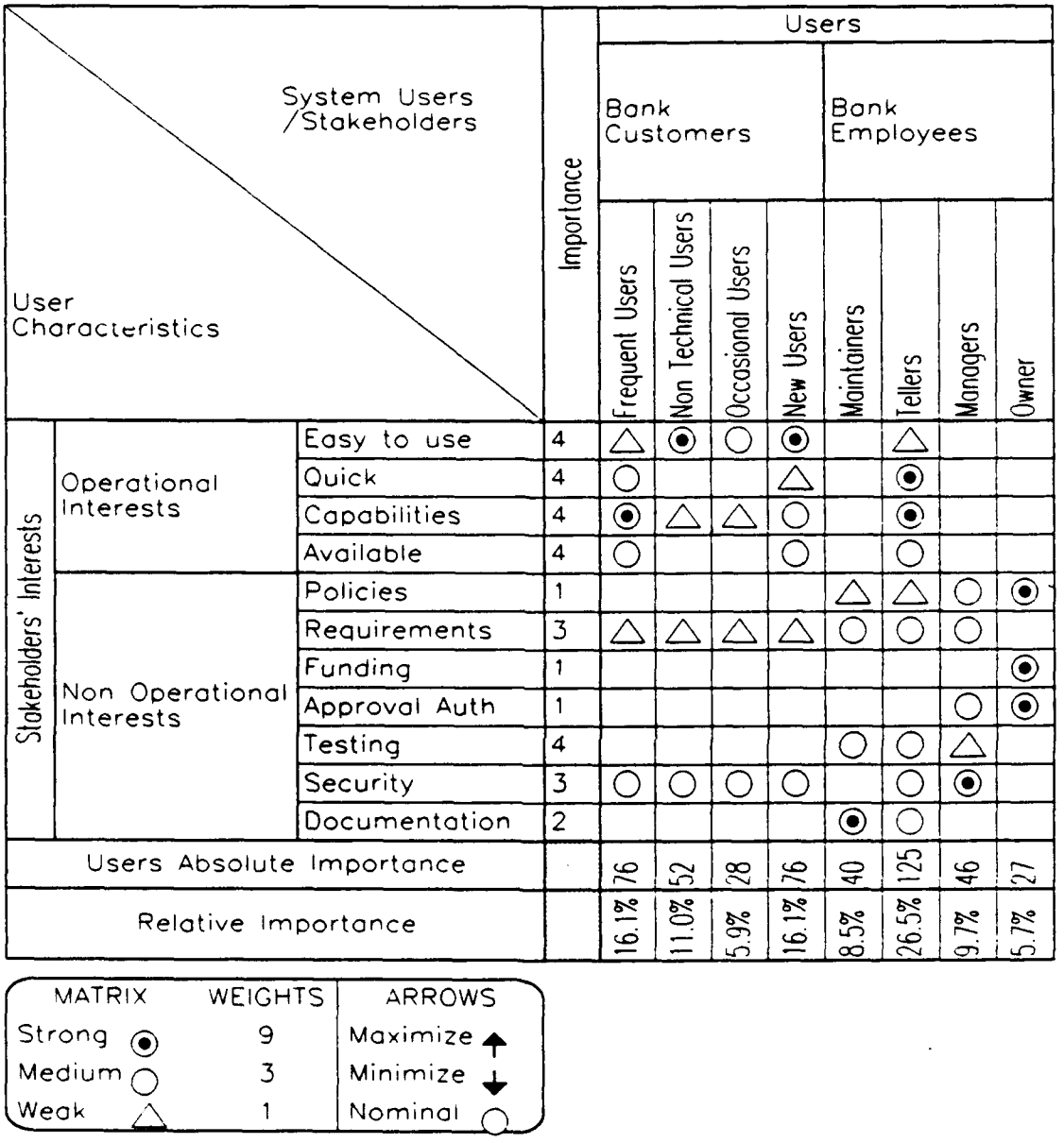


Figure 6. AUTOTELLER Z-0 Matrix User Characteristics vs Users

In our example, the users' interests has been divided into two categories. The first describes operational interests such as: easy to use, quick, capabilities, and available. The other category describes non operational interests such as: policies, funding, and security. These interests may be determined through interviews, surveys, focus groups, trouble reports etc (Zultner, 1990:134). The relationships between the users and the user interests are determined and represented by the symbols above. If no relationship exists, the matrix is left blank. Subjective values of the importance of the users' interests are also included. The final step is to solve the matrix by summing the product of the values

of importance(s) and the relationship(s) and entering the value in the row beneath the matrix. The result is an absolute value of each users' importance or clout. A relative value is calculated for clarity.

This matrix shows that for the AUTOTELLER system, the most important users are the tellers with frequent and new bank customers second.

User Requirements. Now that we have an understanding of the relative importance of the users, we can begin to determine the users' requirements.

With the same techniques as those used above, we collect and review the users' requirements and arrange them in hierarchical fashion on the left side of the Z-1 matrix. We also include the importance values that we gathered from the users. The Z-1 matrix relates the users to the users' requirements in order to place a priority on each user requirement based on the importance of the associated users (Zultner, 1990:135).

		System Users / Stakeholders →	Importance	Stakeholders: Bank Customers: Frequent Users	Bank Customers: Non Technical User	Bank Customers: Occasional Users	Bank Customers: New Users	Bank Employees: Maintainers	Bank Employees: Tellers	Bank Employees: Managers	Bank Employees: Owner	Raw Priority	Adjusted Priority	Relative Adjusted Priority
User Requirements ↓														
Bank Customers' Req	User Interface	Menu System	5	○	◉	○	○					213.3	1066	4.0%
		Friendly Prompts	5		◉	△	◉					249.8	1249	4.7%
		Responsive	5	◉	△	△	○		△			236.6	1183	4.5%
		Trans Receipt	4	○	○	○	○		○			226.8	907	3.4%
	Capabilities	Make Withdrawal	4	◉								144.9	579	2.2%
		Make Deposits	2	○	○	△						87.2	174	0.6%
		Check Balances	3	◉	◉	○	○					309.9	929	3.5%
		Transfer Funds	3	○	△	△	△					81.3	243	0.9%
		Pay Loans	1	○	△							59.3	59	0.2%
Bank Employees' Req	Accounting	Maintain Balances	5						◉			238.5	1192	4.5%
		Update Limits	2	△					△			42.6	85	0.3%
	Information Tracking	Collect Statistics	2					△	○	◉	○	192.4	384	1.4%
		Track Cash	4					○				25.5	102	0.3%
		Track Receipts	2					○				25.5	51	0.1%
		Verify ATM Card	5						○			79.5	397	1.5%
	Misc	Available	5	△				◉	◉			331.1	1655	6.3%
		Secure Accounts	5	○	○	○	○		○	◉	○	331.2	1656	6.3%
		No Comp Errors	5	△	△	△	△		○	◉	△	221.6	1108	4.2%
Users Relative Importance				16.1	11.0	5.9	16.1	8.5	26.5	9.7	5.7			

Figure 7. AUTOTELLER Z-1 Matrix User Requirements vs Users

The requirements were broken down into two groups, those associated with the bank customers, and those associated with the bank employees. Further breakout followed the requirements identified for the sample problem. Once all the requirements were in place, relationships between the user requirements and the users were determined. At the bottom of the matrix, the users' relative importance (multiplied by one hundred) was brought forward from the Z-0 matrix.

Once all the information is in place, the matrix can be solved. The users relative importance (at the bottom of the matrix) is multiplied by the values of the relationships, then added to arrive at the raw priority (on the right hand side of the matrix) of the user requirements. This raw priority is then multiplied by the importance value and the result is the adjusted priority of the user requirements with the user importance taken into account. Relative values are calculated for ease of understanding.

Technical Requirements. Now that the user requirements and their priorities are understood, the technical requirements needed to arrive at those user requirements can be determined. The A-1 matrix is used here. These requirements should be a measurable property of the system which can be manipulated to help meet the user's requirements (Sharkey, 1990:10).

The user requirements were brought forward from the Z-1 matrix and placed down the left hand side of the A-1 matrix. The adjusted priority from the Z-1 matrix (multiplied again by one hundred) is also brought forward. The technical requirements, placed across the top, were broken into two groups. Those related to the user interface and those relating to the internal operations of the ATM. For each technical requirement, a target value is identified. These values should be such that they can be verified at a later time. A direction of improvement may also be added. For example, the statistics should take no more than one second to be updated, the quicker the better. For PIN attempts however, a value of three is assigned, no more or no less than three. This is symbolized by the open circle. Once the technical requirements have been established and reviewed, relationships are filled in using the same symbols as earlier.

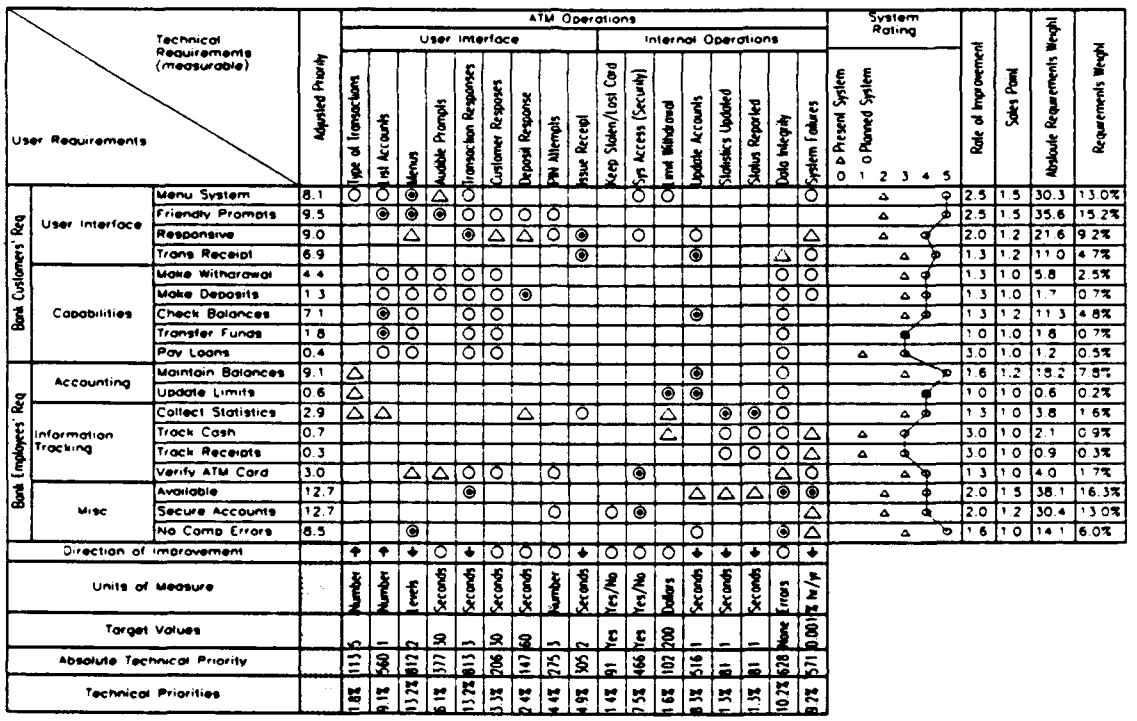


Figure 8. AUTOTELLER A-1 Matrix User Requirements vs Technical Requirements

The A-1 matrix includes an assessment of the present and planned systems. Testing of the present system may be used to assess its position, on a scale of one to five, on each of the user requirements (King, 1989:4-5). If we were in a competitive environment, the competitors' positions on the same user requirements would also be evaluated. This information, combined with the adjusted priority, can help establish a plan for where the improvements to the new system should be emphasized. For example, since the present system only rates a two for friendly prompts, and friendly prompts has a relatively high adjusted priority, we will plan the new system to attain a five in this area. The rate of improvement is simply the ratio of planned system to present system (King, 1989:4-6).

The sales point is intended to apply added emphasis on a feature that could result in higher sales for a commercial product. The sales point can take on values of 1.5, 1.2, or 1.0, and are assigned sparingly (King, 1989:4-6). For the AUTOTELLER system, sales points were assigned to user requirements that were felt would lure additional customers to the ATM.

The absolute requirements weight is determined by taking the product of the adjusted priority, rate of improvement, and sales point for each user requirement (King, 1989:4-6). The requirements weight is simply the percentage of the absolute requirements weight, making it easier to understand.

Now the matrix is solved using the same techniques as before, but this time multiplying the relationships by the absolute requirements weight. The resulting absolute technical priority (and the percentage values), identifies the priorities of the technical requirements based on the users' priorities. This will again be brought forward to the next matrix.

Processes. The next step is to take the technical requirements identified in the A-1 matrix and model them with a software analysis model. If a structured analysis approach is desired, we would use process models and entity models (Zultner, 1990:136). An object oriented approach may use objects and services as a basis for analysis. We will continue with the structured analysis methodology. A different structured analysis could well have resulted at this point. However, for the sake of simplicity, and in order to better draw conclusions of the usefulness of SQFD, we will adopt the same data flow diagrams previously developed for the AUTOTELLER system, thereby maintaining a basis for comparison.

Across the top of the A-2 matrix are the processes identified through our structured analysis. The processes are identified by their name and number. Down the left hand side are the technical requirements and their priorities as determined from the A-1 matrix. Relationships are determined and placed within the matrix as before. When the matrix is solved by multiplying the relationships by the technical importance, we determine an importance value for each process. This now tells us the value to the user of each process in the system. A similar matrix can also be constructed for the data entities in the system, resulting in an importance rating for each data entity.

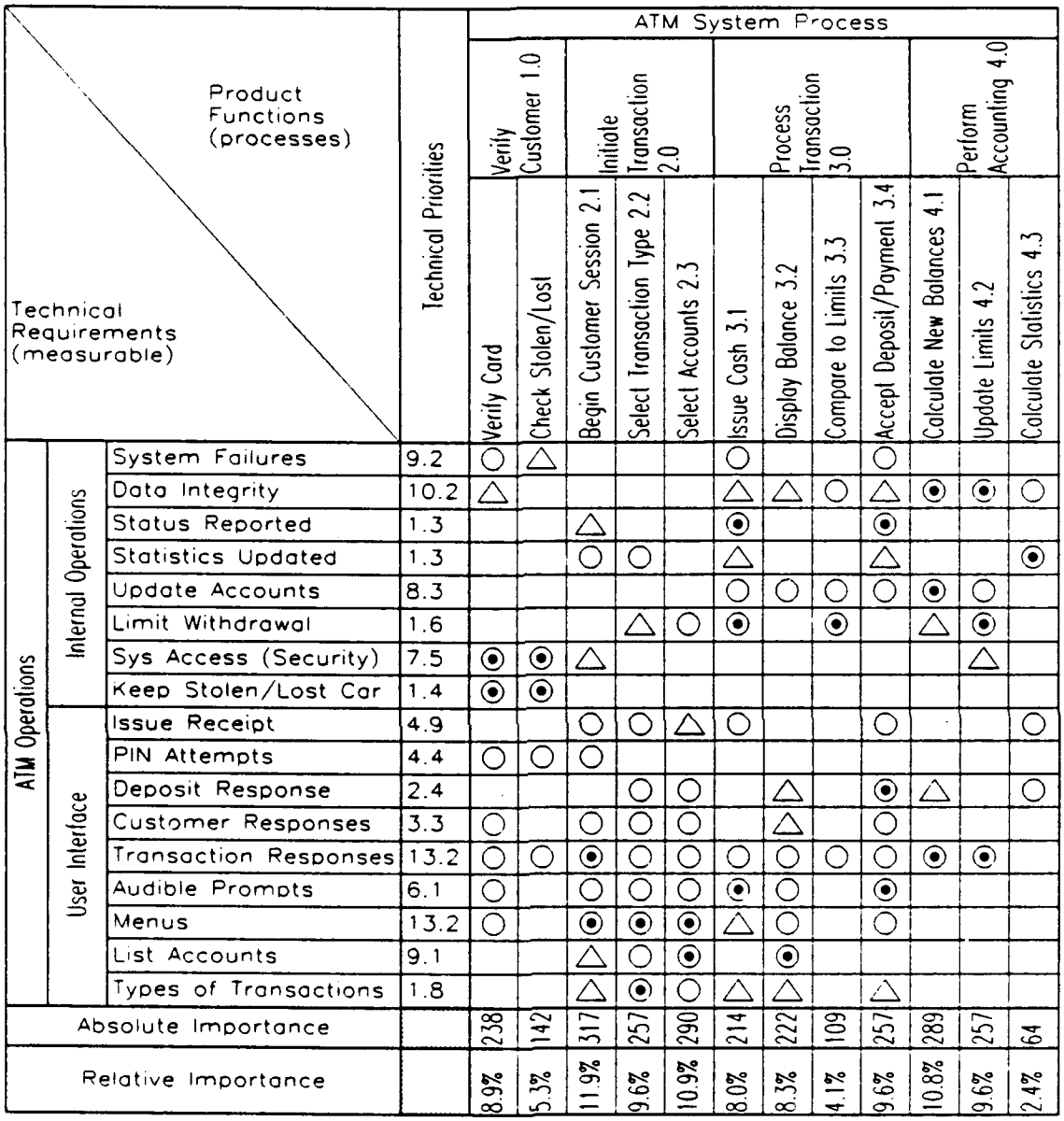


Figure 9. AUTOTELLER A-2 Matrix Technical Requirements vs Product Functions

This is as far as this sample problem will be studied. The SQFD process may be carried out further to study failure modes, new concepts, effort, cost, reliability etc; however, the representation presented here is all that is necessary for our purposes.

Please answer the questions on the following pages. Your informed and honest responses to these questions will help to determine the course of future research projects.

Again, if you have any questions, or need further clarification on some points, feel free to contact me at any time.

References

1. Hauser, John R. and Don Clausing "The House of Quality," *Harvard Business Review, 3*:63-73 (May-June 1988).

2. King, Bob. *Better Designs in Half the Time, Implementing QFD Quality Function Deployment in America* (Third Edition). Methuen MA: GOAL/QPC, 1989.

3. Loy, Patrick H. and James M. Mitchell. "Software Requirements Specification: The AUTOTELLER Automatic Teller System," *Standards, Guidelines, and Examples on Systems and Software Requirements Engineering*, edited by Dorfman, Merlin. and Richard H. Thayer. 439-456. Los Alamitos CA: IEEE Computer Society [illegible]s, 1990.

4. Sharkey, Allen I. "Generalized Approach to Adapting QFD for Software," Presentation for the IBM Corporate Education Center. Thornwood NY, December 1990.

5. Zultner, Richard E. "Software Quality [Function] Deployment," *Transactions from the Second Symposium on Quality Function Deployment.* 132-143. Planning Committee for A Symposium on Quality Function Deployment, 1990.

Software **Quality Function Deployment** Questionnaire

Please complete the questionnaire and return it to the address below NLT 26 Jul 91 or contact me and I will pick it up. Additional pages may be used if necessary. Thank you for your participation.

AFIT/LSG
ATTN Capt Craig Lamb
Bldg 641 WPAFB OH 45433

Name and Rank: _____ Organization: _____

Please choose the area below that most closely describes your function. If none does so adequately, specify on the "other" line:

_____ Software Acquisition Management _____ Software Engineering
_____ Academia _____ User Organization
_____ Other: _____

1. Do you think that the structured analysis representation of the sample problem (not taking into account the SQFD analysis) adequately captures all the users' wants, needs, and desires? Please explain.

2. Does the SQFD approach help to capture any of those needs, wants, desires, you felt were missing (if any) from the Structured Analysis approach? Please explain.

3. Key to the SQFD process is capturing the importance (or priority) of the various characteristics of the problem. Do you believe that this is missing from most Structured Analysis representations? Is this useful information to determine?

4. In your opinion, would the use of SQFD provide a better means of requirements traceability then methods you currently employ? Please explain.

5. The SQFD Z-0 matrix attempts to analyze the different users and their relative importance or clout. From your experience, do you believe that this would be useful information? Please explain.

6. Do you believe that the additional information resulting from using SQFD could result in a better Structured Analysis of the problem? In other words, would SQFD provide a better front end to Structured Analysis then present analysis methods? Please explain.

7. Based on your experience, are there any reasons why SQFD would not be "workable" in the USAF environment? Are there obstructions that would not make SQFD possible to use?

8. If you were trained to use SQFD, would you seek to apply it to a real-world program?

9. Any additional comments you may have would be greatly appreciated, including any suggested modifications to the SQFD process.

Appendix B: Survey Data

The survey was distributed to twenty-two knowledgeable personnel. The breakout of these personnel were as follows:

- four to Academia
- four to software acquisition managers
- ten to software engineers
- and four to user organizations

The breakout of the survey responses were as follows:

- four from Academia
- four from software acquisition managers
- four from software engineers
- and two from user organizations

The raw data was transposed for the sake of legibility and is presented here. The data is arranged with all fourteen respondents answer following the respective question. The respondents are identified by the following acronyms:

AC - #	Academia
SAM - #	Software Acquisition Manager
SE - #	Software Engineer
USER - #	User

A summary of the findings are presented in Chapter V.

1. Do you think that the structured analysis representation of the sample problem (not taking into account the SQFD analysis) adequately captures all the users' wants, needs, and desires? Please explain.

AC - 1: No, it does not capture the non-functional behavioral requirements like timing, performance, security, etc.

AC - 2: No. Two main requirements are [???] neglected: Timing constraints & requirements priorities. The first is essential, the second is important from a development life cycle point of view.

AC - 3: No. DFDs can only model certain types of user requirements

AC - 4: Yes. If the analysis was well performed, all of the requirements were captured. However, if a "want" or a "desire" was not important enough, it should not have been included as a requirement and was probably not.

SAM - 1: Can't tell but I believe so. The SAR approach gives more an implementation of the user's requirements with the interfaces between the applicable systems. It is hard to tell if this adequately will capture all the user's needs w/o some type of requirements traceability matrix.

SAM - 2: I think that for the purpose if this problem, the users' requirements were adequately addressed. From the users' mentality (not meant to put anyone down) it is important to view the system requirements from their perspective (i.e, the ATM user is only interested in his/her time and money - not the bank folks; the bank folks have to be considered with their needs as well as the bank customers' needs) in relation, we have to sit in the pilots seat from time to time to view their perspective.

SAM - 3: No.

SAM - 4: No, lists of flow diagrams in no way give any quantification of relative weight of various requirements relative to user requirements.

SE - 1: No - the structured analysis did not address audible signals, transfer of funds, or the minimum information to be printed on the receipts. It also did not explicitly identify the system as menu driven, although this is implied by "response time for menu changes..." (p.11). And it did not identify the requirement for loan payments, although it is assumed this is satisfied by "Accept Deposit or Payment" (p.10)

SE - 2: No. Not unless you include the constraints/non-behavioral requirements with the DFDs.

SE - 3: Yes, although the quality of the users' wants is dependent on the quality of the user survey.

SE - 4: The structured analysis representation with stepwise refinement is an excellent method of defining and refining user requirements. This assume the user, however, is involved throughout the requirements generation process.

USER - 1: Some attributes seem to be missing - such as Durability of the machine itself. Idiot - proof operations, abuse resistant, understandable prompts/voice etc. The maintainer requirements appear to be left out.

USER - 2: The SA seems to be process oriented by bringing in things like the clock - it gets right into the flow charting before there is a complete look at the problem and solution required.

2. Does the SQFD approach help to capture any of those needs, wants, desires, you felt were missing (if any) from the Structured Analysis approach? Please explain.

AC - 1: Can't answer because you did not provide a complete structured analysis. Where is the data dictionary? In many cases, the DD would contain these non functional requirements. At least some of them anyway.

AC - 2: It does cover both; however, the timing should be more explicit-Graphical techniques are easier to understand than a matrix & therefore less likely to be missed or misinterpreted.

AC - 3: It helps to capture some of them in a matrix form yes. It's not clear how useful that would be for system developers. A mechanism is still required to get the designer(s) to see how these user requirements map to other system representations such as DFDs.

AC - 4: Yes. If there were any differences between a requirement and a desire about a particular function, both the requirement and the desire were probably delineated in the SQFD matrices.

SAM - 1: Again, can't tell for sure. I believe the SQFD is more of a structured approach (i.e. closer to a requirements traceability matrix) and more adequately illustrates the accomplishment of the user's requirements.

SAM - 2: As an ATM user (as a bank customer) I feel the requirements were sufficient.

SAM - 3: Yes.

SAM - 4: Certainly, since the interview and analysis approach captures users view of relative weight of requirements and subsequently importance and priority necessary to sway a design one way or the other. Certainly the two approaches must be handled in tandem/parallel to identify and refine further details and their priority.

SE - 1: It explicitly requirements menus, transfer of funds, and loan payments, but it does not explicitly require audible signals or the minimum receipt information (although it should have, and easily could have, included these missing requirements).

SE - 2: Yes, it is a much more analytical analysis of the needs. The only concern is that the "sample" chosen to represent the desires (and in turn the weighting) is valid.

SE - 3: The SQFD helps document user needs. The need still need to be allocated to components of the detailed (structured analysis) design. SQFD help identify key attributes to be allocated to the design.

SE - 4: The SQFD approach helps identify needs that might not be captured using solely the structured analysis approach.

USER - 1: The maintenance/support functions do not appear to be addressed, unless that all falls under "Available."

USER - 2: SQFD seems to analyze the problem while Structured Analysis seems to start with assumption that the problem is well known. The relative priority assignment is a good step.

3. Key to the SQFD process is capturing the importance (or priority) of the various characteristics of the problem. Do you believe that this is missing from most Structured Analysis representations? Is this useful information to determine?

AC - 1: In some cases yes but not always. It depends in many cases on the modelers domain experience and SA experience. With SQFD its at least explicitly called for & not left up to modelers experience & it is useful info.

AC - 2: Yes, it is missing from most representations. In our development environment (i.e. government), the budget is greatly influenced by outside factors out of the control of the program manager. In the case of potential budget cuts & schedule slippages, choices need to be made to reduce the requirements to meet these budget & schedule constraints. A priority list is essential for that purpose.

AC - 3: Yes, it is missing and yes, it is useful.

AC - 4: Yes. Yes. It is missing and it is useful info to have.

SAM - 1: Yes this is missing from SAR & it is useful. Most users I work with are willing to sacrifice lesser requirements in order to keep the more important requirements.

SAM - 2: In our line of work - yes, we do not have the relative priority of many of our user requirements at the beginning of the conceptual design. Only after design and during test do the priorities surface.

SAM - 3: Yes, it is useful information from a prioritization of functions to develop, test, etc.. If the characteristic is a "real requirement" then its priority is a moot

point from a delivered product standpoint. In other words, if the product is required to have certain characteristics then they are all the same priority.

SAM - 4: Yes & Yes

However, the users change their minds; especially when their representatives change over mid stream in the development process.

SE - 1: The priority information is missing from the Structured Analysis. This information would be very useful during requirements definition when some requirements may have to be deleted or traded off due to resource (memory, timing, size, weight, power, heat, etc) or financial constraints.

SE - 2: Yes. Yes, its critical.

SE - 3: Yes, this is always useful in assessing the viability of a design, determining the salient characteristic to measure from a design.

SE - 4: Yes, this is very critical in understanding the design requirements. W/O this understanding, system design (and ultimately future changes) may cause latent errors W/O a full appreciation for the reason for the requirements.

USER - 1: Most appears to be present and it appears to be useful.

USER - 2: I guess that the SA process implicitly treats the importance whereas the SQFD is explicit. Because it is implicit, it can be missed.

4. In your opinion, would the use of SQFD provide a better means of requirements traceability then methods you currently employ? Please explain.

AC - 1: Maybe slightly better than some of the other matrixing methods.

AC - 2: Not applicable to our academic environment but it is better than the standard structural analysis technique.

AC - 3: Possibly but not necessarily. From an academic point of view, it could help an organization to better define system requirements and how to trace them. But if an organization already had good traceability techniques, SQFD would just help them understand the relative importance of the requirements better.

AC - 4: Yes. Current methods use several matrices in several documents to provide requirements traceability. In most cases this info is inherent in the complete single set of SQFD matrices.

SAM - 1: Yes, SQFD would translate the user's requirements from the SORD or ORD to something more substantial in a PIDS - Of course as derived requirements begin to exist in the lower echelons of development, it may be possible to do SQFD for each layer.

SAM - 2: The concept of SQFD should first be applied to our operational requirements, and yes, traceability of requirements would be better especially when the users' personal tend to change which causes the priorities to change throughout the development.

SAM - 3: No, but it will point out which customers would be the most/least satisfied with the resulting implementation.

SAM - 4: Certainly, since it documents priorities along with details of the design.

SE - 1: I do not believe it would enhance traceability that much, but it would identify why requirements were deleted. It might also enhance verification (testability).

SE - 2: Yes. It would provide the analysis which in turn provides the rationale for your design.

SE - 3: Only with an allocation to the design. As always designs are assessed and managed at box, WBS level. Design parameters must be tracked at this level. SQFD will allow user requirement to be better satisfied assuming they can be allocated for design and recombined to one system requirements.

SE - 4: Yes - but only if implemented up-front (early-on) in the program. Inserting this process midstream would only serve to exaggerate [the] deficiencies long ago accepted by the user.

USER - 1: Better traceability of relative importance of requirements than what we currently do. We have no prioritization of known requirements.

USER - 2: Do not employ any. Most users won't so they can't make any judgement of whats best - in fact this approach could quite confuse them. I'm not sure it would give them confidence that their needs were being considered unless a good training session and program is instituted.

5. The SQFD Z-0 matrix attempts to analyze the different users and their relative importance or clout. From your experience, do you believe that this would be useful information? Please explain.

AC - 1: I'm not sure yet. However, they probably would take issue with their rating if they ever found out you considered them less important than some other user.

AC - 2: Yes, from the standpoint of #3 above. This method results in a much more accurate picture of the requirements priorities.

AC - 3: This could be useful if a system is being developed incrementally to determine which capabilities should be completed first. It also can help system testers to determine the relative effort to be expended in testing various requirements. Of course, it could also set off arguments (about who is most important) and controversy!

AC - 4: Yes. It would be very useful to know & understand the different opinions & desires of the several users about each requirement. This info could be used to appropriately decide if a tougher "requirement" (actually a desire) is worth pursuing or fulfilling.

SAM - 1: Possibly. You still need one central focal point within the USER to identify the strongest requirements. It helps to get input from all users but a unified response and single focal point is important for the person/people implementing the user's requirements.

SAM - 2: In this instance it seems proper. However, in the DOD environment, the users making the final decisions are not always the actual users but several steps above. This tends to distort the actual needs and the clout they bring.

SAM - 3: Yes, it would help differentiating between customer requirements and desirements.

SAM - 4: Certainly, if the "important users with clout" would not change faces so often.

SE - 1: In a commercial system, designed to attract customers, this may be true. But, in a military system, there are often users and requirements that would have to be given an infinite weighting since they are absolute requirements. For other requirements, the relationships may be useful, but I do not believe that you could always say that for example, a user with one strong (=9) relationship equals a user with nine weak (=1) relationships to the requirement.

SE - 2: Yes, same as above.

SE - 3: Yes, the training and capabilities of user determine an appropriate design.

SE - 4: Yes, to differentiate between real requirements and nice to have wants.

USER - 1: I think so. It should at least identify where requirements came from and who was pushing them.

USER - 2: I believe that selection of stakeholder's interests could be arbitrary & thus not show the correct importance. The importance of the stakeholder's interests should be given & worked into the model. Assuming that it is correct, how will it be used? I am not sure that the ratio of customers to employees represents a balance. Someone can arbitrarily skew the importance column if there are more columns for the employees for example then customers. Its the customers that make the bank profitable, not the employees so I would recommend a means of checks & balances on the distribution of columns vertically & also horizontally.

6. Do you believe that the additional information resulting from using SQFD could result in a better Structured Analysis of the problem? In other words, would SQFD provide a better front end to Structured Analysis then present analysis methods? Please explain.

AC - 1: Can't tell from this "pedagogical" example.

AC - 2: Yes, it would since the two deficiencies pointed out are resolved. A graphical timing representation is still needed.

AC - 3: The only way I can see this helping SA is by making the analysts think the system through better, thus helping them develop a better system because they understand it better. But I see no other connection.

AC - 4: Yes. Yes. It would help understand what the different user's operators, maintainers etc perspectives are of the systems capabilities & requirements.

SAM - 1: Yes. SQFD could "bridge the gap" to SAR; its hard to see how all SAR requirements are captured in design.

SAM - 2: The present methods are not bad but could always use improvement as all processes do at some point or another. The SQFD could provide a different maybe more in depth analysis of the problem and hence, could add to current methods.

SAM - 3: No.

SAM - 4: Obviously, as a basis for design and design changes down stream, this is appropriate.

SE - 1: I believe that SQFD would ensure that no important requirements are missing (not reflected in) th structured analysis. The structured analysis would have a higher probability of being complete.

SE - 2: Don't know. Haven't worked with structured analysis enough.

SE - 3: Yes, it aids in user requirements definition. A process many time poorly completed if at all before design.

SE - 4: This additional information would aid in the definition process up front - reducing need for creeping requirements and system deficiencies.

USER - 1: My experience is limited, but this approach appears useful.

USER - 2: There was no analysis done for the C-29 before the contractor started work. If we as the user had not worked with them it could have been a disaster, so I think SQFD is a good start and that combining it with SA would be a big step.

7. Based on your experience, are there any reasons why SQFD would not be "workable" in the USAF environment? Are there obstructions that would not make SQFD possible to use?

AC - 1: No.

AC - 2: Other than ensuring that the "user" (receiving command) is not confused with the "government developer" (i.e. AF Sys Cmd) I see no problem.

AC - 3: Like SA, it's another one of those approaches that would work well when a system is well understood from the beginning, and particularly if it's not very complex. It looks like it would become very difficult and cumbersome to use for large, complex problems.

AC - 4: Possibly. In the current procurement process there are no "convenient" ways to ask a contractor to perform the SQFD analysis. Additionally, there are very few AF people knowledgeable enough to help in the SQFD analysis. (But it is possible to do.)

SAM - 1: Other than training and the perception of a new procedure and the limitation imposed by society to "change", no. Would there be additional time/resources needed to use SQFD, however.

SAM - 2: I think SQFD could be applied to any environment. Key word "applied." Every problem is different. The main obstruction that would make SQFD difficult would be changes.

SAM - 3: No. Yes, the contractors must be convinced of its superiority to their current practices. We specify functional and performance requirements as a customer and do not "tell" the contractors how to provide the capability.

SAM - 4: There is no reason, it could not be used as a tool amongst a host of others to supplement this process.

SE - 1: "Absolute" priority of some requirements/users. Often our users do not weight their requirements. The prefer to give us an overall set of requirements and have us do all of them. Also, on large programs, it would be difficult to accurately weight all requirements and do the matrices.

SE - 2: 1. No.

2. Possibly because it doesn't take into account the opinion of "experienced" peoples opinions.

SE - 3: Yes, but it should be used early to avoid invalidating mature design.

SE - 4: As mentioned in 4, this process must be installed at the time of program initiation (such as an ATF or NASP program.) It will serve no useful purpose in a program that's been around for 10 yrs.

USER - 1: This approach could be de-railed by the committee approach to design. That often produces a product that sort-of-works for everyone but not very well for anyone.

USER - 2: Computerizing a person's job (like the C-29) takes much thought & planning, particularly to identify a way to put code to things that have not been defined precisely. I think SQFD is a good start but there should be something preceding it to insure that the matrix is properly completed.

8. If you were trained to use SQFD, would you seek to apply it to a real-world program?

AC - 1: Yes, even though it seems to be a better config management & resource allocation tool than a "technical" front-end for SA.

AC - 2: Couldn't really make the choice without more in depth info.

AC - 3: The ideas look very useful, but I doubt that I would use it in the form presented. No matter how formal you make the method, it is still based on subjective opinions. Hence, the result is subjective indicators of importance. I may do just as well by assigning these values directly to the final matrix.

AC - 4: Yes. Most definitely.

SAM - 1: A simple program at first in order to understand the ins and outs.

SAM - 2: Yes.

SAM - 3: No. If I were convinced of its benefits I would try to convince counterparts in industry.

SAM - 4: Certainly

SE - 1: I could probably use it in a program, such as C-29 has been recently, that is in an ECP mode, where a list of user requirements/enhancements is being pared down to fit within available funding. On an initial program, unless automated tools and user cooperation were available, it would take too long and probably be ignored.

SE - 2: Yes.

SE - 3: Yes, operator displays and controls are a key candidate for this analysis.

SE - 4: Yes, under the constraints of 4 and 7.

USER - 1: I would give it a try.

USER - 2: If I understood it a bit more.

9. Any additional comments you may have would be greatly appreciated, including any suggested modifications to the SQFD process.

AC - 1: How do you handle large problems? Is there a hierarchical process for this?

AC - 2: What about a responsibility matrix since users #'s are very large & requirements come from a variety of sources. This falls in line with TQM concepts.

AC - 3:

Pros:	Cons:
- a nice formal process.	- results may be so subjective as to
- produces useful information.	not justify the effort required.
- makes users think concepts	- needs to be integrated with the
through more thoroughly.	method used to be really useful.
- can help to direct where most	- appears to become very complex
effort should be expended and	and cumbersome as complexity
what priorities should be used.	of the problem (system) increases
	(in other words, it's not clear that it
	would scale up well.)

AC - 4: Survey was difficult to answer - it was very subjective. SQFD (and QFD in general) would be of great value to today's system development.

SAM - 1: I think SQFD would be great but I'm not sure if I could totally understand or explain the use of it. I also have a fear that this would be another great idea

that is shelved because of the complexity of the tool and the education that would be required at all levels in order to implement it.

SAM - 2: None.

SAM - 3: In the acquisition business we specify functions and performance. Contractors propose methodologies to develop software and we evaluate/select contractors based upon our perception of the best methodologies. Therefore, we need education of the different methodologies to make informed evaluations. Adoption of certain methodologies can only be achieved through convincing industry that the methodology is superior to others.

SAM - 4: No

SE - 1: - Allow for "absolute" requirements/users.

- In Matrix A-1, how did you determine the system rating for the planned system? This appears to be the one calculation that is subjective and could distort the final priorities. If an objective method was used, please give me more details.

SE - 2: None.

SE - 3: Ways to allocate requirement to components or ways to determine if requirement can be allocated to components or can only be leveled at a system level would be useful.

SE - 4: Requirements allocation is a very critical (yet often over-simplified) process that requires constant management and persistent traceability.

USER - 1: None.

USER - 2: I see it as a method of identifying such things as requirements but would be cautious in using any numbers that result from the matrix, reference my comments in Q1 & 5. The data flow diagrams are good & appear to be what Sierra used - however we didn't understand them when presented because we had no training.

Appendix C: AUTOTELLER Software Requirements Specification

Software Requirements Specification:
The AUTOTELLER Automatic Teller System

Patrick H. Loy
Loy Consulting, Inc.
3553 Chesterfield Avenue
Baltimore, Maryland 21213
(301) 483-3532

and

James M. Mitchell
80 Fiek Avenue
Toksook Bay, Alaska 99637
(907) 427-7113
(508) 757-5540

Representing
Tundra Software, Inc.*

* This is a fictious software engineering company.

Table of Contents

1: Introduction

1.1: Purpose

The purpose of this document is to present in a precise and easily understood manner all software requirements deemed necessary upon a formal review by this office (Tundra Software, Incorporated [TSI]) and its customer, the Bank of CSUS. It will satisfy the functional, performance, interface, design, and verification requirements for the computer software to be developed as part of the automatic teller machine (ATM) system described in the "System Requirements Specifications." This document is intended to be a baseline to supply sufficient design information to the Bank of CSUS as a foundation for software assessment and approval. It provides TSI with a basis for software design.

1.2: Scope

The objective of this project is to describe the software requirements of the ATM system in accordance with the product specifications stated in the "Systems Requirements Specifications" document. The deliverable product will be referred to as *TELLERFAST*[2] and will be a package that includes the following:

- *Software System:* This will be a software product used on a microcomputer with at least a 160-Megabyte mass-storage device and with an interface to the bank's current database. It will control the operation of the ATM, provide prompts to the customer, do the necessary accounting to produce the written report of the transaction to the customer, maintain a record of all services performed, and perform accounting to update the bank's database.

- *Software documentation:* Complete and easily understood documentation of the software will be provided to aid in future maintenance and/or modification of the software.

- *Operations manual:* Used by the bank personnel, it will explain everything necessary to operate and maintain the software system and the ATM.

- *User's manual:* A guide to be handed out to all ATM users. It will explain, step-by-step, exactly how to use the ATM. It will include diagrams to illustrate the steps to be performed and the parts of the ATM.

1.3: Definitions, Acronyms, and Abbreviations

ACCOUNT DESIGNATION – Designation that the customer has given his account to identify it; may be a date, or word such as Xmas, etc.

ATM – The automatic teller machine (hardware)

ATM CARD – Plastic card, issued to the customer by the Bank, that enables the customer to use the ATM

ATM CARD NUMBER – Number on the ATM card that identifies the customer

AUTOTELLER – The computer system composed of *TELLERFAST* software and the ATM

[2] The TELLERFAST package will not provide any software services other than those specified in this document.

BANK OF CSUS – The Bank of California State University at Sacramento

BYTE – Unit of memory storage in the machine needed to store one character of information

COMPUTER SYSTEM – Computer equipment and programs that accomplish a set of objectives

CUSTOMER – The bank's patrons

CUSTOMER IDENTIFICATION NUMBER – Permanent number in the bank's database identifying the customer (distinct from the ATM card number)

DATABASE – Information relating to the bank's customers and their financial status

DATA DICTIONARY – Defines the contents of the data flows shown in the data-flow diagrams

DATA FLOW DIAGRAM – Shows inputs and outputs to, from, and between the processes and external elements of the system

DESIGNER – Tundra Software, Incorporated, the developer of AUTOTELLER and *TELLERFAST*

DFD – Data-flow diagram

HIS – In the interest of brevity, all uses of his/her have been eliminated to simply "his"

K – Unit of memory storage in a computer, 1K = 1024 bytes

MB, MEGABYTE – Unit of memory in a computer; 1MB = 1 million bytes

MENU – What the customer sees on the ATM screen; a list of choices from which the customer selects a banking activity

MENU DRIVEN – Refers to the TELLERFAST software; using a sequence of prompts and menu responses to lead the customer through the use of AUTOTELLER

MODULAR PROGRAMMING – Construction of programs in sections, each of which performs a single function

PIN – Personal identification number; the number associated with the customer ID to identify legal use of the ATM card

PROCESS – Unit of work performed by the ATM

PROCESS SPECS – Detailed specification of a process shown in a DFD

PROMPT – A message displayed on a screen requesting a response from the user

SERVICE LIMITS – There are 2 types of Service Limits:

- Bank Limits – Maximum and minimum dollar amounts set by the bank for the use of the ATM
- Individual Withdrawal Limits – Maximum a customer is allowed to withdraw from an account based on account balances and the amount already withdrawn on current date

SESSION – Time and actions occurring between the insertion of the ATM card and the issuing of the receipt

STANDARD WAIT TIME – Maximum time allowed for the customer to respond to a prompt

STATUS INQUIRY – Inquiry made by bank personnel about the state of the ATM. (is it operational? How much cash does it have in stock etc.?)

SRS – Software requirement specifications

SYS – System requirements specifications document

TELLERFAST – Software (program) for AUTOTELLER

TRACEABLE – A requirement in this document can be traced back to the corresponding requirement in the SYS document that makes this software requirement necessary

TSI – Tundra Software, Incorporated

USER – There are two types of users: the bank personnel, and the bank's customers

USER FRIENDLY – Computer system using software that is so simple to operate that it can be used by untrained users

1.4: Document Overview

This document has three major sections and two appendices:

1. Section 1 (Introduction) provides an overview of the entire SRS document.

2. Section 2 describes the product that will be produced.
   It includes:

   - Product perspective
   - Product activity
   - User characteristics
   - General constraints
   - Assumptions and dependences

3. Section 3 addresses the specific requirements of the TELLERFAST system.
   It includes:

   - Functional requirements – These include inputs, process specs, and outputs for each primitive process in the data-flow diagrams. The inputs and outputs are direct references to the data in the data dictionary.
   - External interface requirements
   - Performance requirements
   - Design constraints
   - Attributes
   - Other requirements

4. Appendix A contains the data dictionary

5. Appendix B contains the data-flow diagrams (DFDs)

2: General Characteristics

2.1: Introduction

This section introduces the software product. It describes the characteristics and limits affecting the product and its requirements.

2.2: Product Perspective

2.2.1: The *TELLERFAST* software package will perform as a part of the ATM system described in the system requirements specifications of the Bank of CSUS. This software product will provide the control necessary for the ATM system to perform its activities.

2.2.2: The initialization of the *TELLERFAST* software will be performed by the Bank of CSUS as described in Section 3. As the controlling software for the ATM system, the *TELLERFAST* software interfaces with the local system hardware devices. These devices are the card reader, cash dispenser, keypad, display screen, printer, and vault gate.

2.3: Product Activity

The TELLERFAST software will perform the following activities:

- Verify customer identification
- Select service
- Deposit cash or check
- Withdraw cash
- Pay to loan account or selected credit card
- Transfer funds between the customer's accounts
- Inquire of customers' account balances
- General accounting

2.4: User Characteristics

2.4.1: We assume that the customers will be occasional users without any background and training in computers. We also assume that the customer can read at a forth grade level, and can follow simple instructions during the ATM's operation.

2.4.2: We assume that the bank personnel will receive training in the use of *TELLERFAST*. These users will be provided with all reference materials, instructions, and documentation for the ATM.

2.5: General Constraints

The following are general design constraints for *TELLERFAST*:

- Data encoding scheme: ASCII character set as defined by the ANSI X3.4-1977 standard.
- Total available memory for programming logic, tables, etc., as specified in this document shall not be exceeded.
- The use of a microcomputer with a minimum of one MB of memory.

- The program will be written in COBOL.
- TSI will adhere to the bank's accepted accounting practices as listed in U.S. Banking Stat. 0073.
- The customer will be guided through the use of the AUTOTELLER services with easy-to-read instructions.

2.6: Assumptions and Dependencies

2.6.1: The bank of CSUS has a central computer system that can interact with AUTOTELLER during execution of *TELLERFAST*.

2.6.2: The bank's database will be accessible in real time.

2.6.3: The bank's database will be modified to identify customer accounts as legitimate ATM accounts.

2.6.4: The bank's database will be changed to meet *TELLERFAST* 's requirements. The PIN, the ATM card number, and the service limits will be added to the existing information. This change will be made prior to developing the *TELLERFAST* software.

3: Specific Requirements

3.1: Functional Requirements

This section, organized in subsections, contains the details necessary for the systems engineer to create the design specifications of the ATM.

3.1.1: Verify Customer ID (DFD Process 1.0)

3.1.1.1: Introduction

"Verify Customer ID" will read the customer's ATM card number from his ATM card and will prompt the customer for his PIN. This function will verify that the customer is authorized to use the ATM.

3.1.1.2: Inputs

- ATM-card-number
- ATM cust-id-info
- PIN
- Lost-stolen-list

3.1.1.3: Processing

Upon insertion of the customer's ATM card
Check the ATM-card-number against the Lost-stolen-list
If the entered ATM-card-number is on the list
Then
Retain card and suspend processing

If the entered ATM-card-number is not on the Lost-stolen-list then
Repeat the following:
Prompt customer for his PIN
Compare the entered PIN to the PIN listed in the Bank's database for this customer
Until
the customer enters his correct PIN or
fails to correctly enter his PIN after 3 attempts

If a valid PIN has been entered
Then
Continue processing
Else
Suspend processing
Issue message prompt

3.1.1.4: Outputs

- Cust-id-number
- Prompts to enter PIN
- Message prompt (to include the reason processing is suspended)

3.1.2: Initiate Transaction (DFD Process 2.0)

3.1.2.1: Begin Customer Session (DFD Process 2.1)

3.1.2.1.1: Introduction

"Begin Customer Session" assembles and prints the Receipt-header-info.

3.1.2.1.2: Inputs

- Cust-ID-number
- Date-time

3.1.2.1.3: Processing

Read the date-time from the system clock
Direct the printer to print receipt-header-info on customer-receipt

3.1.2.1.4: Outputs

- Receipt-header-info

3.1.2.2: Select Transaction Type (DFD Process 2.2)

3.1.2.2.1: Introduction

"Select Transaction Type" enables the bank customer to designate the type of transaction desired by using the ATM terminal keys to select from a menu.

3.1.2.2.2: Inputs

- Cust-ID-number
- Trans-type

3.1.2.2.3: Processing

Repeat the following:
Display the transaction menu
Prompt the customer for trans-type
If the trans-type is not terminate-session then
Issue trans-type to process 2.3
Until
the customer terminates the session, or
4 transactions have been processed during this customer session
Issue eject-receipt-signal

3.1.2.2.4: Outputs

- Trans-type
- Eject-receipt-signal

3.1.2.3: Select Accts (DFD Process 2.3)

3.1.2.3.1: Introduction

"Select Accounts" enables the bank customer to designate the account(s) to be used in a transaction. The customer's selection is made by pressing ATM terminal keys in response to a menu display.

3.1.2.3.2: Inputs

- Acct-selection
- Cust-acct-list
- Trans-type

3.1.2.3.3: Processing

Display menu of customer accounts from cust-acct-list. Depending upon the trans-type, prompt the customer for either one or two account selections, as indicated in the decision Table below.

CONDITION: Trans-type	DECISION RULE				
	1	2	3	4	5
Balance Inquiry	Y	N	N	N	N
Deposit	N	Y	N	N	N
Fund Transfer	N	N	Y	N	N
Loan Payment	N	N	N	Y	N
Withdrawal	N	N	N	N	Y
ACTION					
Prompt for Source Account			X		X
Prompt for Destination Account		X	X	X	
Prompt for Balance Inquiry Account	X				

KEY: Y = True
N = Not True
X = Take Action

3.1.2.3.4: Outputs

- Prompts
- Service-info

3.1.3: Process Transaction (DFD Process 3.0)

3.1.3.1: Issue Cash (DFD Process 3.1)

3.1.3.1.1: Introduction

"Issue Cash" dispenses cash to the customer for the valid-amt requested.

3.1.3.1.2: Inputs

- Valid-amt

3.1.3.1.3: Processing

If there are sufficient bills in the cash dispenser
    Then
        Issue service-amt to cash dispenser control
    Else
        Display error message
        Terminate the session
        Notify operations of system error condition
        Put affected ATM system on standby

3.1.3.1.4: Outputs

- Service-amt

3.1.3.2: Display Balance (DFD Process 3.2)

3.1.3.2.1: Introduction

"Display Balance" displays the balance of the account selected by the customer.

3.1.3.2.2: Inputs

- Service-info
- Balance

3.1.3.2.3: Processing

For each acct-number
    Display acct-designation and balance

3.1.3.2.4: Outputs

- Balance-display

3.1.3.3: Compare Amount Requested with Limits (DFD Process 3.3)

3.1.3.3.1: Introduction

"Compare Amount Requested with Limits" verifies that the service-amount is within the limits of the valid account.

3.1.3.3.2: Inputs

- Date-time
- Withdrawal-info
- Balance-limits
- Service-info
- Requested-trans-amt

3.1.3.3.3: Processing

If date-last-withdrawn is not current date
    Then
        Change date-last-withdrawn to current date
        Change daily-withdrawal limit to $200
Prompt customer for requested-trans-amt
Based on the appropriate trans-type and acct-selection given
    If the applicable condition is true
        1. Requested-trans-amt > max-deposit-allowed
        2. Requested-trans-amt > credit-card-limit
        3. Requested-trans-amt > checking/savings account balance
        4. Requested-trans-amt > daily-withdrawal-limit
        5. Requested-trans-amt > acct-withdrawal-limit
    Then
        Display an error message and terminate the session
    Else
        Issue valid-service-info or valid-amt
        Issue trans-info

3.1.3.3.4: Outputs

- Prompts
- Trans-info
- Valid-service-info
- Valid-amt

3.1.3.4: Accept Deposit or Payment (DFD Process 3.4)

3.1.3.4.1: Introduction

"Accept Deposit or Payment" opens the depository slot and accepts the customer's deposit or payment envelope.

3.1.3.4.2: Inputs

- Envelope-accepted
- Valid-service-info

3.1.3.4.3: Processing

Issue open-slot to the depository
Prompt the customer to enter the envelope
Upon receipt of envelope-accepted
    Continue processing

3.1.3.4.4: Outputs

- Open-slot

3.1.4: Perform Accounting (DFD Process 4.0)

3.1.4.1: Calculate New Balances (DFD Process 4.1)

3.1.4.1.1: Introduction

"Calculate New Balances" calculates and updates the new balances of the customer's accounts, after the selected transaction is over.

3.1.4.1.2: Inputs

- Date-time
- Trans-info

3.1.4.1.3: Processing

For each customer session, obtain operation-complete and trans-info from process 3.0. Depending upon the trans-type update the account balances, as indicated in the decision Table below:

CONDITION: Trans-type	DECISION RULE				
	1	2	3	4	5
Balance Inquiry	Y	N	N	N	N
Deposit	N	Y	N	N	N
Fund Transfer	N	N	Y	N	N
Loan Payment	N	N	N	Y	N
Withdrawal	N	N	N	N	Y
ACTION					

No Action	X			
Subtract Requested-trans-amt from Balance1		X	X	X
Add Requested-trans-amt to Balance1	X			
Add Requested-trans-amt to Balance2		X		

KEY: Y = True
N = Not True
X = Take Action

3.1.4.1.4: Outputs

- Receipt-details
- Trans-details
- Audit-details

3.1.4.2: Update Limits (DFD Process 4.2)

3.1.4.2.1: Introduction

"Update Limits" update the daily-withdrawal-limit of the customer to reflect the amount of cash withdrawn during the current transaction.

3.1.4.2.2: Inputs

- Trans-info

3.1.4.2.3: Processing

Subtract requested-trans-amt from daily-withdrawal-limit to obtain updated-limits.

3.1.4.2.4: Outputs

- Updated-limits

3.1.4.3: Calculate Work Statistics (DFD Process 4.3)

3.1.4.3.1: Introduction

"Calculate Work Statistics" maintains statistical information regarding activity at a particular ATM machine. It accumulates the following running totals:

- Number of customer sessions
- Number of transactions completed
- Number of each transaction type completed

It also maintains status information on the following:

- Amount of cash-left
- Number of customer-receipts remaining in dispenser
- Dollar amount expected in the depository vault

3.1.4.3.2: Inputs

- Trans-details

3.1.4.3.3: Processing

For each customer session,
Do the following to obtain ATM-statistical-info:
Add 1 to the customer sessions total
Subtract 1 from the customer-receipts supply
For each transaction:
Add 1 to the total for the appropriate transaction type
If the trans-type is withdrawal
Then
Subtract requested-trans-amt from cash-left
If trans-type is deposit
Then
Add requested-amt to deposits-expected total

Reset totals as follows:

If the cash supply is replenished
Then
Reset cash-left to the amount remaining
If the customer-receipts supply is replenished
Then
Reset the customer-receipts supply to the amount remaining
If the depository vault is cleared of deposits received
Then
Reset the deposits-expected to zero
If the running activity totals are read from the bank database
Then

Reset them to zero

3.1.4.3.4: Outputs

- ATM-statistical-info

3.2: External Interface Requirements

3.2.1: User Interfaces

The *TELLERFAST* software will have the following users:

- The bank customer for personal banking transactions
- The bank personnel for service, inquiry, and maintenance

3.2.1.1: Bank Customer

3.2.1.1.1: General ATM Operation

The customer initiates an ATM session by inserting his ATM card into the card reader slot. The customer enters his choice of transaction and requested amounts for the transactions by pressing keypad buttons on the front of the ATM console. The ATM will display lighted messages to guide the customer through a transaction. If the customer makes a mistake, an appropriate message will be displayed. One customer receipt will be printed for an ATM session. On deposit transactions, the customer is instructed to insert a deposit envelope. The customer receives a cash withdrawal from the cash dispenser slot.

3.2.1.1.2: Customer Receipt

The customer receipt (see Figure 1) will contain this information:
Bank name
Account number
Date
Time of day
Description of the type of transaction
Automatic teller location code number
Account balance at end of session

Bank of CSUS	
Date	Time
March 17, 1989	09 44
ATM Teller #9	
Account Number	459 612 340
Deposit to Checking	$150.00
Account Balance:	$2,537.49

Figure 1: Sample Customer Receipt

3.2.1.2: Bank Personnel

3.2.1.2.1: General ATM Operation

Bank personnel acting as customers will interact with the ATM in a similar manner to typical bank clientele, as described in Section 3.2.1.1.

3.2.1.2.2: Internal Operations

Bank personnel interaction with AUTOTELLER, the ATM and *TELLERFAST* is described in the *ATM Users Manual* (Doc. #201-4108; TSI). It details the complete use and operation of AUTOTELLER from the bank teller's perspective, including customer use, status inquiry, and troubleshooting.

3.2.2: Hardware Interfaces

The *TELLERFAST* software will interface with the following hardware:

- The bank's main computer
- The secondary storage device containing the bank's database
- The ATM

3.2.3: Software Interfaces

The *TELLERFAST* software will interface with the following software:

- The bank's database, to access and update the information in the customer's accounts and to add to the transaction
- The bank's computer system, to respond to status inquiries

3.3: Performance Requirements

3.3.1: The response time for menu changes will be no more than three (3) seconds.

3.3.2: The response time when a file is accessed will be no more than five (5) seconds.

3.3.3: The time to read an ATM card will be no more than three (3) seconds.

3.3.4: The time to issue a printed receipt after the customer has terminated a session will be no more than two (2) seconds.

3.3.5: The maximum time allowed for customer response will be thirty (30) seconds, with the exception of envelope insertion, which must be performed within sixty (60) seconds.

3.3.6: When AUTOTELLER is used, the beeper will sound until the customer responds to the menu prompt or for thirty (30) seconds, whichever comes first.

3.4: Design Constraints

3.4.1: The software will be menu driven. Menu selection will be made by pressing the ATM keypad keys. Numerical entries will be made by using a ten key pad.

3.4.2: Prompts will be displayed before user makes each entry.

3.4.3: Confirmation of action taken, input accepted, or error condition will be displayed after each input.

3.4.4: Error messages will be displayed after the detection of a system fault and the system will respond only to status inquiries by authorized bank personnel.

3.5: Attributes

3.5.1: Introduction

AUTOTELLER will possess certain quality attributes built into the work product specified in this document.

3.5.1.1: Reliability

TELLERFAST will have been thoroughly tested at time of delivery so that computational errors will not occur. *TELLERFAST* will be written in a modular structure to make modification as easy as possible.

3.5.1.2: Maintainability

TELLERFAST will be maintained by Bank of CSUS. System and software documentation will be supplied by TSI that will provide bank employees quick instruction on use and maintenance of the system software. Modules will be as independent as possible so that changes in one module will not

produce software errors in another part of the system. TSI will be available for consultation regarding maintenance.

3.5.1.3: Software Security

In addition to the electronic protection schemes that will be introduced, there will also be the following:

- Customers of Bank of CSUS will have magnetically encoded cards to insert for proper identification, as well as a personal code for a system password and verification.
- The AUTOTELLER database will be encrypted using standard practices and measures.
- Restrictions will be placed on bank personnel and anyone who has electronic access to the records of the system to insure data integrity and compliance with government privacy codes.

3.5.1.4: Robustness

AUTOTELLER will continue to function accurately despite incorrect customer input. Under no circumstances will the system "lock up" or fail because of user error. Stringent error checking procedures will be in effect during all operations of the machine and at all levels of system design and code.

3.5.1.5: Availability

AUTOTELLER will be out of operation for service no more than .001% of its yearly operating time, in addition to its regularly scheduled preventative maintenance.

3.6: Other Requirements

3.6.1: Transaction File

AUTOTELLER will maintain a file that contains a record of each transaction that occurs at the automated teller during a 24-hour period.

The information retained on each transaction will include:

- Customer identification
- Account(s) affected
- Type of transaction
- Amount of the transaction
- Beginning balance(s)

- Ending balance(s)
- Date, time, and location

The file will be accessible to the bank's main computer system in accordance with the bank's established security-access procedures.

3.6.2: Monitoring of ATM Status

Certain information about the status of the ATM will be available online to authorized bank personnel. Such information will include:

- The amount of cash dispensed
- The amount on hand
- The number of customer receipts on hand

In the event of a mechanical failure in the ATM, AUTOTELLER will send a signal to the bank's main computer system console. The type of failure will be handled by the system hardware in the form of a non-fatal system interrupt, explaining both the existing error condition and the module location of the failure.

Appendix A: Data Dictionary

Acct-designation	= 1{alphanumeric-character}15
Acct-details	= audit-details + ATM-statistical-info + trans-details
Acct-selection	= [savings/checking/loan-acct/credit card] + (acct-designation)
Acct-withdrawal-limit	= amount *usually initialized at $200 each day*
Acct-number	= 8{numeric-character}8
Amount	= dollars + cents
ATM-card-number	= 7{numeric-character}7
ATM-cust-id-info	= PIN + ATM-card-number + cust-id-number
ATM-statistical-info	= 1{trans-type + valid-amt + time + date}4
Audit-details	= completed-trans-info + time + date + location + cust-id-number
Balance-display	= *display of current balance to customer*
Balance-Limits	= 1{balance}2 + service-limits
Balance	= amount *funds available in an account*
Balance1	= amount *used in transactions involving 2 accounts*
Balance2	= amount *used in transactions involving 2 accounts*
Cash-dispensed	= 1{twenty dollar bills}10
Cash-left	= amount *used in tracking the ATM cash supply*
Completed-trans-info	= 1{previous-balance + valid-amt + updated-limits + new-balance + 1{acct-number}2}2
Credit-card limit	= amount
Cust-acct-list	= 1{acct-designation + acct-number}8
Cust-id-number	= 8{numeric-character}8

Cust-info	= cust-card-PIN + requested-trans + requested-trans-amt
Customer-receipts	= 1{numeric-character}4 *used to keep track of the supply of blank receipts available*
Daily-withdrawal-limit	= amount
Date	= 6{numeric-character}6 *month/day/year*
Date-last-withdrawn	= date
Date-time	= date + time
Deposits-expected	= {numeric-character}
Eject-receipt-signal	= *signal to the printer that the customer session is complete and the receipt should be printed*
Envelope-accepted	= *signal from the depository hardware*
Location	= 5{alphanumeric-character}5 *terminal identification*
Lost-stolen-list	= {ATM-cardnumber}
Max-deposit-allowed	= amount
Money-checks	= (cash) + (checks)
New-balance	= amount
Open-slot	= *signal to the depository hardware*
Operation-complete	= *message to the accounting process*
PIN	= 4{numeric-character}4*personal id number*
Previous-balance	= amount
Prompts	= *messages displayed on the terminal screen*
Receipt-details	= 1{completed-trans-info}4
Receipt-header-info	= time + date + location + cust-id-number
Requested-trans-amount	= amount
Requested-trans	= [trans-type/acct-selection]

Service	= [cash-dispensed/deposit-accepted/loan-payment-accepted/balance-displayed]
Service-amt	= amount
Service-info	= trans-type + 1{acct-number}2
Service-limits	= (daily-withdrawal-limit) + (max-deposit-allowed) (credit-card-limit)
Time	= *"hour/minute"*
Trans-details	= 1{new-balance}2 + trans-type
Trans-info	= 1{previous-balance + valid-amt + trans-type + service-limits + 1{acct-number}2}2
Trans-type	= [deposit/withdrawal/balance inquiry/transfer funds/loan payment/terminate-session]
Updated-limits	= service limits *after a transaction*
Valid-amt	= amount *validated against the database*
Valid-service-info	= *service-info that has been validated with the database*
Withdrawal-info	= date-last-withdrawn + acct-withdrawal limit

Appendix B: Data Flow Diagrams

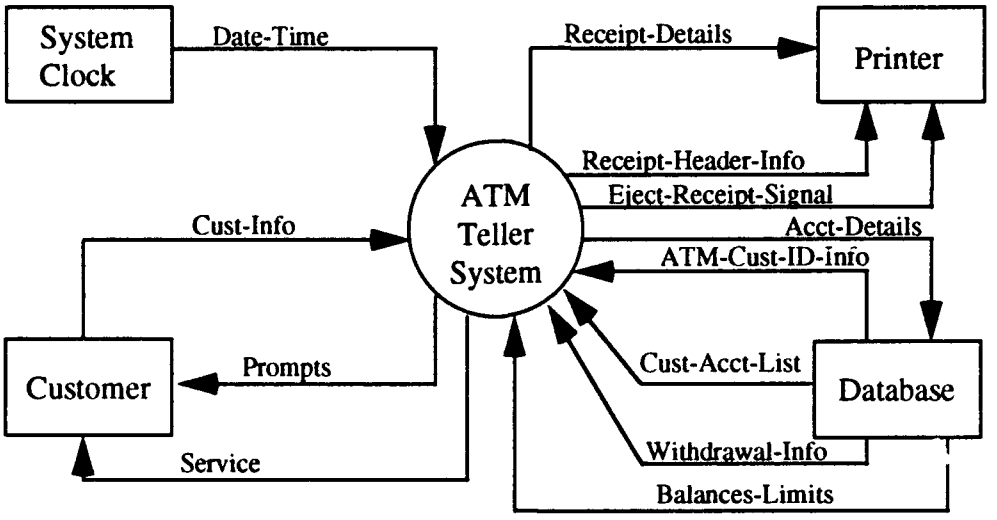


Figure 2: Context Diagram

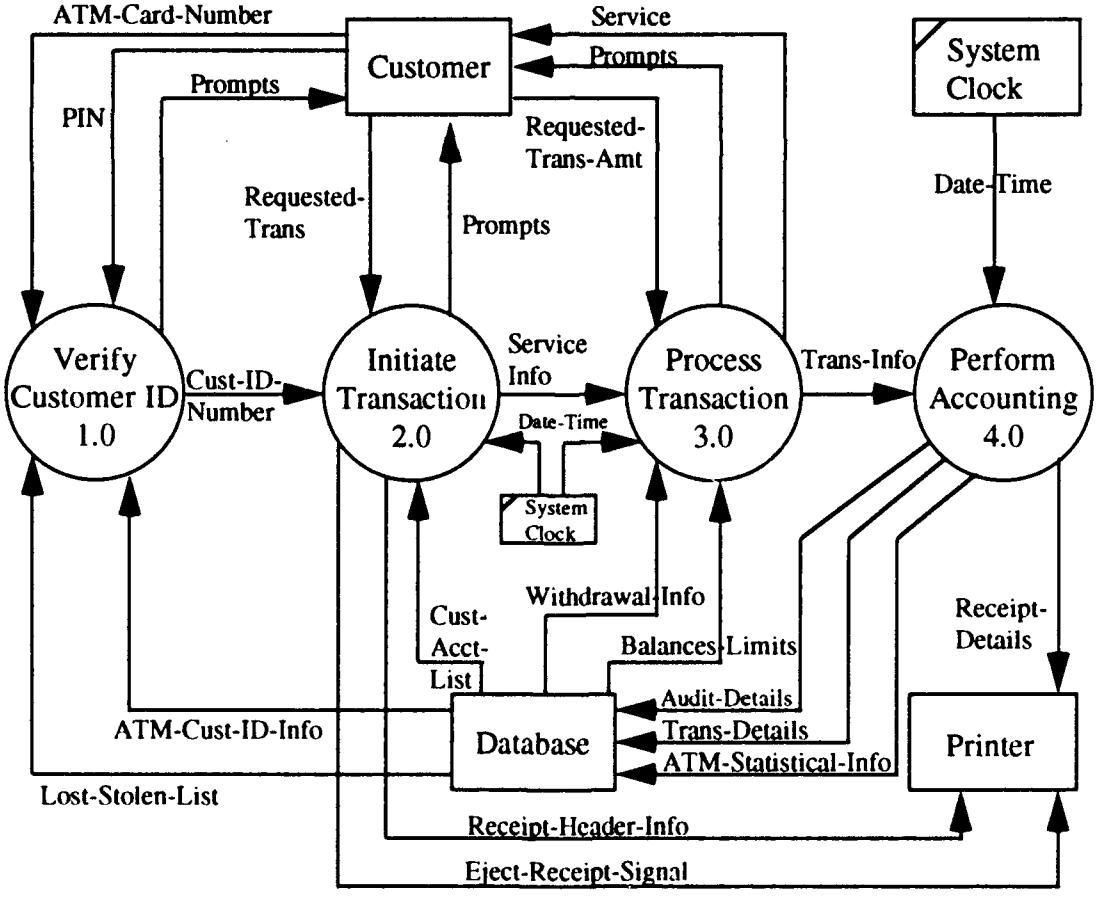


Figure 3: Top Level Diagram

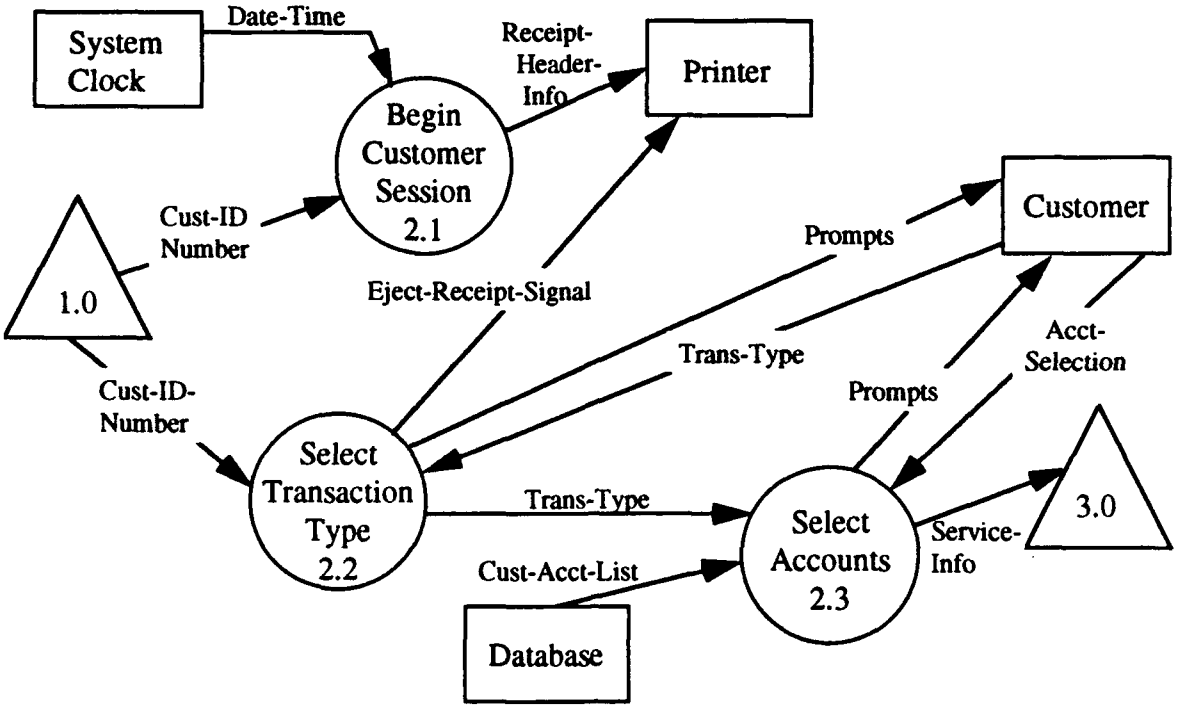


Figure 4: Process 2.0 Initiate Transaction

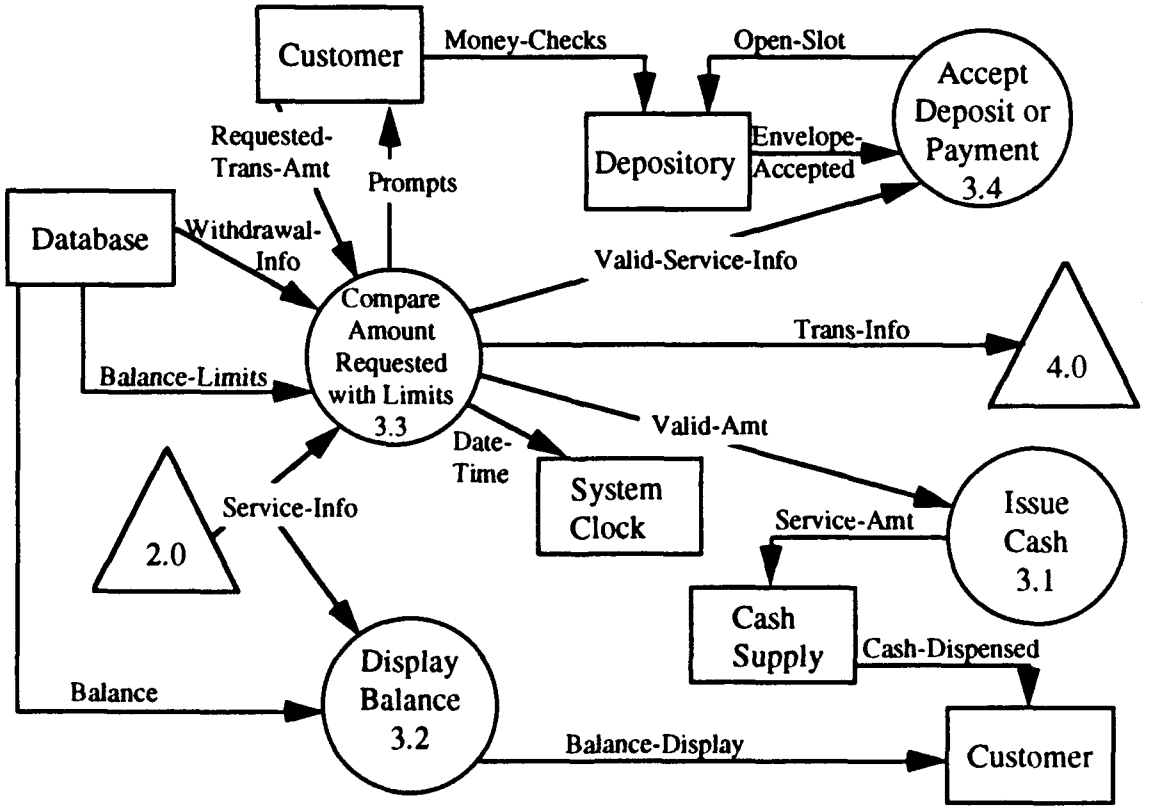


Figure 5: Process 3.0 Process Transaction

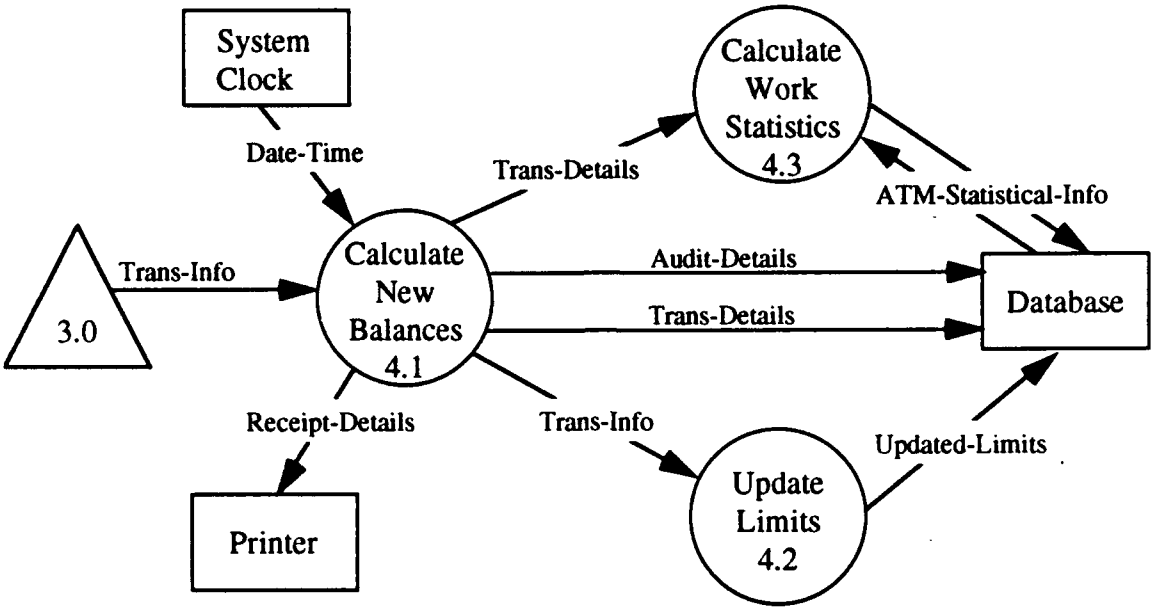


Figure 6: Process 4.0 Perform Accounting

Bibliography

Akao, Yoji. *Quality Function Deployment.* Cambridge MA: Productivity Press, 1990.

Brooks, Frederick P. Jr. "No Silver Bullet," *IEEE Computer, 20*:11,17 (19 April 1987).

Brown, Susan. QFD Coordinator. Telephone discussion. American Telephone & Telegraph, 25 April 1991.

Canan, James W. "The Software Crisis," *Air Force Magazine, 69*:46-52 (May 1986).

Committee on Science, Space, and Technology. *Bugs in the Program, Problems in Federal Government Computer Software Development and Regulation.* Staff Study by the Subcommittee on Investigations and Oversight, 101st Congress, 1st Session, September 1989. Washington: Government Printing Office, 1989.

Conti, Tito. "Process Management and Quality Function Deployment," *Quality Progress, 22*:45-48 (December 1989).

Davis, Alan M. *Software Requirements Analysis and Specification.* Englewood Cliffs NJ: Prentice Hall, 1990a.

-----. "The Analysis and Specification of Systems and Software Requirements," *Systems and Software Requirements Engineering,* edited by Thayer, Richard H. and Merlin Dorfman. 119-144. Los Alamitos CA: IEEE Computer Society Press, 1990b.

DeMarco, Tom. *Structured Analysis and System Specification.* Prentice Hall, Englewood Cliffs NJ: Prentice Hall, 1978.

De Vera, Dennis and others. "An Automotive Case Study," *Quality Progress, 21*:35-38 (June 1988).

Department of the Air Force. *Operational Needs, Requirements, and Concepts.* AFR 57-1. Washington: HQ USAF, 7 October 1988.

Eureka, William E. and Nancy E. Ryan. *The Customer-Driven Company.* Dearborn MI: ASI Press, 1988.

GOAL/QPC. *Customer Guide.* Methuen MA: GOAL/QPC, undated.

Hauser, John R. and Don Clausing. "The House of Quality," *Harvard Business Review, 3*:63-73 (May-June 1988).

King, Bob. *Better Designs in Half the Time, Implementing QFD Quality Function Deployment in America* (Third Edition). Methuen MA: GOAL/QPC, 1989.

Kitfield, James. "Is Software DOD's Achilles' Heel?," *Military Forum, 5*:28-35 (12 September 1989).

Loy, Patrick H. and James M. Mitchell. "Software Requirements Specification: The AUTOTELLER Automatic Teller System," *Standards, Guidelines, and Examples on Systems and Software Requirements Engineering*, edited by Dorfman, Merlin and Richard H. Thayer. 439-456. Los Alamitos CA: IEEE Computer Society Press, 1990.

Martin, James and Carma McClure. *Structured Techniques for Computing*. Englewood Cliffs NJ: Prentice Hall, 1985.

Myers, Ware. "Software Pivotal to Strategic Defense," *IEEE Computer, 22*:92-97 (January 1989).

Peters, Lawrence. *Advanced Structured Analysis and Design*. Englewood Cliffs NJ: Prentice Hall, 1987.

Porter, Bob. "Implementing QFD at TI: What Worked and What Didn't," *Transactions from the Symposium on Quality Function Deployment*. 305-335. Dearborn MI: ASI Press, 1989.

Pressman, Roger S. *Software Engineering: A Practitioner's Approach* (Second Edition). New York: McGraw-Hill Book Company, 1987.

QualiSoft Corporation, QFD Designer 2.0 Demonstration file. West Bloomfield MI: QualiSoft Corporation, 9 April 1991.

Richards, Evelyn. *Pentagon Finds High-Tech Projects Hard to Manage*. The Washington Post, 11 December 1990, page 1 col 1.

Shaikh, Khushroobanu I. "Thrill Your Customer, be a Winner," *Transactions from the Symposium on Quality Function Deployment*. 287-303. Dearborn MI: ASI Press, 1989.

Sharkey, Allen I. "Generalized Approach to Adapting QFD for Software," Presentation for the IBM Corporate Education Center. Thornwood NY, December 1990.

Sommerville, Ian. *Software Engineering* (Third Edition). Wokingham England: Addison-Wesly Publishing Company, 1989.

Stewart, Nick P. "Software Error Costs," *Quality Progress, 21*:48-49 (November 1988).

Sullivan, Lawrence P. "Quality Function Deployment," *Quality Progress, 19*:39-50 (June 1986).

Thompson, Dianne M. M. and M. Hosein Fallah. "QFD - A systematic Approach to Product Definition," *Transactions from the Symposium on Quality Function Deployment.* 277-285. Dearborn MI: ASI Press, 1989.

Yeh, Raymond T. and Peter A. Ng. "Software Requirements - A Management Perspective," *Systems and Software Requirements Engineering,* edited by Thayer, Richard H. and Merlin Dorfman. 450-461. Los Alamitos CA: IEEE Computer Society Press, 1990.

Yourdon, Edward and Larry L. Constantine. *Structured Design.* Englewood Cliffs NJ: Prentice Hall, 1979.

Zultner, Richard E. "Software Quality [Function] Deployment," *Transactions from the Second Symposium on Quality Function Deployment.* 132-143. Dearborn MI: ASI Press, 1990.

Vita

Captain Craig R. Lamb was born 8 August 1963 in Washington DC. He graduated from McLean High School, McLean Virginia in 1981 and later that year matriculated at the Virginia Military Institute. He graduated from VMI in May of 1985 and received the degree of Bachelor of Science in Electrical Engineering. Captain Lamb began his career in the United States Air Force in October of 1985 at Wright-Patterson Air Force Base. His first assignment was in the Aeronautical Systems Division, Deputy for Tactical Systems as the F-4 Modernization Demonstration Project Officer and was subsequently assigned to the Air Defense Aircraft Procurement (ADAP) Program as a Source Selection Administrative Officer. Upon completion of the ADAP source selection, he was assigned as Technical Manager for the Peace Pearl Foreign Military Sales Program. Captain Lamb then transferred to the Deputy for Transports where he became Project Manager for the C-29A program. In addition to these duties, he was also Program Manager for the C-20A/B/D aircraft residual tasks. Upon completion of these residual tasks, Captain Lamb assumed the duty of C-29A Deputy Program Manager. In May 1990 he was assigned to the Air Force Institute of Technology enrolled in the initial Graduate Software Systems Management curriculum. His next assignment will be to the F-22 Systems Program Office.

Permanent Address: 890 Pimlico Dr Apt 3-B
Centerville, OH 45459

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE
POSSIBLE APPLICATION OF QUALITY FUNCTION DEPLOYMENT IN SOFTWARE SYSTEMS DEVELOPMENT IN THE UNITED STATES AIR FORCE

5. FUNDING NUMBERS

6. AUTHOR(S)
Craig R. Lamb, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSS/LSY/91D-8

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objectives of this thesis were to determine whether the methods of Quality Function Deployment (QFD) could be used in the software development environment within the USAF, and whether or not this area should be researched further. The research was limited to the requirements analysis and definition phase.

The different areas of study included a brief review of the structured analysis methodology, a detailed review of the QFD models currently being used in the product industries, a review of how QFD fits into the software development cycle, and specific software modifications to the QFD methodology. A review of some applications of software QFD (SQFD) is also performed. A sample problem consisting of an Automated Teller Machine is developed in detail using the SQFD methods identified.

A subjective survey was conducted of a small sample of USAF software experts to determine the suitability of SQFD for USAF use based on the sample problem.

The results of the thesis show that QFD can be adapted to software development. SQFD also shows potential to save both time and money for the USAF. Consideration must be given to the sample size and nature of the survey when interpreting the results of this research.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Software Engineering, Requirements, Computer Programs, Management, Systems Management, Quality, Quality Assurance, Computer Program Documentation	188
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

AFIT Control Number AFIT/GSS/LSY/91D-8

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years _____ $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

_____ _____

Name and Grade Organization

_____ _____

Position or Title Address